

Mémoire

Présenté à L'Université de Montpellier II en vue de l'obtention de l'

Habilitation à Diriger des Recherches

en Mathématiques appliquées et applications des mathématiques par

Ivorra Benjamin

Professeur Assistant ('Ayudante Doctor') à l'Université Complutense de Madrid
(Espagne)

Méthodes et techniques de Modélisation, de Simulation et d'Optimisation appliquées à divers problèmes industriels

Soutenue le 07 Février 2013 devant le Jury composé de Messieurs:

Nom et Prénom	Grade	Etablissement
Jacques Henry	Directeur de Recherche Émérite	IMB, Bordeaux
Mohamed Masmoudi	Professeur des Universités	IMT, Toulouse
François-Xavier Le Dimet	Professeur des Universités Émérite	Université Joseph Fourier
Alain Rapaport	Directeur de Recherche	INRA Montpellier
Bijan Mohammadi	Professeur des Universités	CERFACS / UM2
Hedy Attouch	Professeur des Universités	Université Montpellier 2

Rapporteurs:

Jacques Henry	Directeur de Recherche Émérite	IMB, Bordeaux
Roland Glowinski	Professeur des Universités	University of Houston
Jesús Ildefonso Díaz Díaz	Professeur des Universités	Univ. Complutense Madrid

Á Emma, Tatiana, Ángel, ma famille et mes amis...

Remerciements

Le travail présenté dans ce mémoire représente plus de six années de ma vie. Durante cette période, de nombreux (et heureux) changements ont eu lieu. J'ai reçu continuellement le soutien de beaucoup de personnes (famille, amis et collègues) et cela a permis le bon déroulement de mon développement scientifique. Je veux leur exprimer toute ma gratitude le long de cette page.

Je tiens tout d'abord à remercier Roland Glowinski, Jesús Ildefonso Díaz Díaz et Jacques Henry d'avoir accepté d'être mes rapporteurs. J'apprécie grandement leur effort et suis conscient du temps dédié à cette tâche.

Je remercie également Jacques Henry, Mohamed Masmoudi, François-Xavier Le Dimet, Alain Rapaport, Bijan Mohammadi et Hedy Attouch de me faire l'honneur de participer au jury de soutenance.

Je tiens aussi à remercier Bijan Mohammadi, Hedy Attouch, Lionel Thibault, Gérard Michaille et Rémi Carles pour m'avoir encouragé et aidé à soutenir cette habilitation. Un grand merci aussi à Bernadette Lacan pour son appui administratif reçu depuis la thèse jusqu'à aujourd'hui.

Les thèmes de recherche développés dans ce document ont été réalisés après ma thèse à l'Université Complutense de Madrid et au sein du groupe de recherche MOMAT qui a su m'accueillir depuis 2006. En particulier, je tiens à présenter toute ma gratitude à Angel Manuel Ramos qui a toujours cru en moi, m'a donné les moyens scientifiques, moraux et financiers de développer ma recherche et m'a inclus dès le début dans de nombreux projets passionnants. Je veux aussi saluer tous mes collègues de la Complutense: Jose-Maria Rey, Juan-Antonio Infante, Sixto Álvarez, Jesús Díaz, Gregorio Diaz, Nadia Smith, Beatriz Martínez López, José Manuel Sénchez-Vizacino, Antonio Brú, Gerardo Oleaga, Luis Bravo, Maria Ruiz, les membres du projet QUIMAPRES et tous les autres. Grâce à eux j'ai gagné non seulement des collègues de travail mais aussi des amis!

Il est aussi important de saluer tous mes collègues d'autres universités qui ont travaillé avec moi sur divers thèmes de recherche ou ont montré de l'intérêt pour mes travaux: (Chili) Miguel Carrasco, Pedro Gajardo, Héctor Ramírez, Felipe Alvarez, Rafael Correa, Roberto Cominetti; (France) Alain Rapaport, Jérôme Harmand; (Espagne) Juana López Redondo, Pilar Martínez Ortigoza; (Mexique) Susana Gómez, Andres Fragueta; (USA) Roland Glowinski, Juan Santiago.

Bien sûr, un grand travail de recherche ne peut se faire sans le soutien et les moments passés avec les amis: François, Sèb, Ségo, Anaïs, Lisa, Lio, Mag, Giulia, Nath, Kuuu, Titi, Miguel, Rafa, Fernanda, Elly, Damien, le groupe friki-mates (un gros bisou à Laura) et la pandilla ex-Neo-Asturiana.

L'appui moral et affectif de ma famille a aussi été une grande source de motivation pour continuer à aller de l'avant: Papa, Maman, mon frère, Marie-Hélène, Romain, Sylvane et toute ma famille française et espagnole: Gaude, Avelina y toda la manada!

Pour terminer, je voudrai remercier ma femme Tatiana et ma fille Emma qui savent m'apporter tout leur amour quand les moments sont difficiles et rendent ma vie tellement plus belle et savoureuse!

Contents

1	Introduction	1
1.1	Problématique générale	1
1.2	Rappels sur les travaux de thèse	2
1.3	Approfondissement des thèmes étudiés en thèse	6
1.4	Nouvelles thématiques post-doctorales	9
2	Optimization Methods	15
2.1	Introduction	15
2.2	General optimization method	16
2.2.1	General description of the MSA	16
2.3	MSA implementations	19
2.3.1	MSA implementation with first order descent core optimization algorithms	19
2.3.2	MSA implementation with second order descent core optimization algorithms	20
2.3.3	MSA implementation with genetic core optimization algorithms	22
2.4	Numerical validation of the method	25
2.4.1	Parameters in algorithms	26
2.4.2	Results	27
2.5	Conclusions and Perspectives	30
3	Shape Optimization with Geometric Parameterization	31
3.1	2D/3D modeling and optimization applied to the design of a fast hydrodynamic focusing microfluidic mixer for protein folding.	32
3.1.1	Introduction	32
3.1.2	Microfluidic mixer modeling	33
3.1.3	Numerical Experiments	37
3.1.4	Numerical Results	41
3.1.5	Conclusions and Perspectives	51
3.2	Shape optimization of geotextile tubes for sandy beach protection	52
3.2.1	Introduction	52
3.2.2	Geotextile tubes and erosion process	52
3.2.3	Flow model	55
3.2.4	Cost function	57
3.2.5	Application to beach erosion control	58

3.2.6	Conclusion	63
3.3	Shape optimization of a dispersive Bioreactor	64
3.3.1	Introduction	64
3.3.2	Mathematical model	65
3.3.3	Optimization problem	68
3.3.4	Results	69
3.3.5	Conclusions and Perspectives	70
3.4	Design of Code Division Multiple Access Filters Based on Sampled Fiber Bragg Grating by Using a Shape Optimization Approach	71
3.4.1	Introduction	71
3.4.2	CDMA filter design problem	73
3.4.3	Numerical test	76
3.4.4	Conclusions and Perspectives	82
4	An Epidemiological model for studying Classical Swine Fever Virus spread between and within farms.	83
4.1	Introduction	83
4.2	Characteristics of Classical Swine Fever	85
4.2.1	CSF evolution	86
4.2.2	Routes of transmission	86
4.2.3	Control measures	87
4.3	Mathematical formulation of the model	87
4.3.1	General description	87
4.3.2	Farms and transports of pigs inputs	90
4.3.3	Within-farm CSFV spread	90
4.3.4	Between-farm CSFV spread	92
4.3.5	State transition due to direct contacts	92
4.3.6	State transition due to indirect contacts	93
4.3.7	State transition due to the CSF natural evolution	95
4.3.8	Detection of contaminated farms	95
4.3.9	Control measures	95
4.3.10	Movement restrictions	95
4.3.11	Zoning	96
4.3.12	Depopulation	96
4.3.13	Tracing	96
4.4	Model Validation	97
4.4.1	Numerical experiments	97
4.4.2	Farms and pig transports inputs	97
4.4.3	Scenarios parameters	97
4.4.4	Considered outputs	98
4.4.5	Comparison with other models	99
4.4.6	Comparison with real epidemic data	101
4.4.7	Results	101
4.5	Analysis of the model robustness	106
4.5.1	Numerical experiments	108

4.5.2	Results	110
4.6	Conclusions and Perspectives	118
5	Modeling and simulation in food engineering	121
5.1	Modeling and simulation of High-Pressure/Temperature processes and inactivation of enzymes	122
5.1.1	Introduction	122
5.1.2	Mathematical model for enzymatic inactivation	122
5.1.3	Heat and Mass Transfer Modelling	124
5.1.4	Coupling of Enzymatic Inactivation and Heat–Mass Transfer Models	135
5.1.5	Conclusions and Perspectives	142
5.2	Estimation of the dynamics of protein folding processes from free energy computations	144
5.2.1	Introduction	144
5.2.2	Some preliminaries on molecular modeling	145
5.2.3	PDE model	148
5.2.4	Estimation of the diffusivity profile	152
5.2.5	Results	153
5.2.6	Conclusions and Perspectives	155
6	Modeling, Simulation and Optimization of a Polluted Water Pumping Process in the Open Sea	159
6.1	Introduction	159
6.2	A mathematical model for oil spills	160
6.3	Modelling of the pumping process	162
6.4	Numerical approximation	163
6.5	Parallel Bi-Conjugate Gradient type method	165
6.5.1	Bi-CG and Bi-CGSTAB methods	165
6.5.2	Parallelization of the Bi-CGSTAB	167
6.6	Numerical analysis of the model	168
6.6.1	Parameters	168
6.6.2	Results	171
6.6.3	Comparison between implicit and explicit schemes	171
6.6.4	Parallelization schemes	172
6.6.5	Influence of the trajectory	174
6.7	Optimal trajectory	175
6.7.1	Problem definition	175
6.7.2	Numerical experiments	178
6.8	Conclusions and Perspectives	184
7	A Variance-Expected Compliance Model for the Optimization of Structure Submitted to Random Loads	187
7.1	Optimization of truss structures submitted to random loads	187
7.1.1	Introduction	187
7.1.2	Standard Minimum Compliance Truss Design	188

7.1.3	Variance-Expected Compliance Model	191
7.1.4	Numerical Study	193
7.1.5	Conclusions	202
7.2	Topology optimization of structure submitted to random loads	202
7.2.1	Introduction	202
7.2.2	Variance-expected compliance approach for topology optimization	204
7.2.3	2D Numerical example	207
7.2.4	3D Numerical example	210
7.2.5	Conclusions and Perspectives	211
8	Conclusions Générales	213
	Bibliography	215

Chapter 1

Introduction

1.1 Problématique générale

La modélisation [51, 56, 67], la simulation [100, 202, 227] et l'optimisation [6, 124, 210] sont trois branches fondamentales et complémentaires des mathématiques appliquées. Elles englobent un large éventail de méthodes et techniques, aussi bien théoriques que numériques, pouvant être directement appliquées afin de résoudre certains problèmes posés par diverses industries (par exemple pétrolière [66], financière [136], ingénierie civile [7] ou sanitaire [8]).

Dans ce contexte, considérons un industriel posant la question générique suivante: "Je possède un certain produit dont je veux améliorer le fonctionnement. Quelle solution les mathématiques peuvent-elles me fournir?". Comme nous le verrons par la suite dans ce document au travers de divers exemples concrets, la marche à suivre pour essayer d'y répondre suit en général le schéma décrit ci-dessous [26, 29]:

- **Modélisation:** La première étape consiste à comprendre et à quantifier les mécanismes induits par le produit. Nous avons pour cela besoin d'outils de modélisation. En particulier, il nous faut choisir les phénomènes à modéliser et ceux à négliger pour des raisons de simplification. Ensuite, nous devons étudier quelle méthode de modélisation (telles que stochastique [109] ou à base d'équation différentielles [34, 68]) est la plus adaptée à notre problème et définir les équations, les paramètres et les sorties gouvernant notre modèle.
- **Simulation:** Une fois la modèle défini, nous devons nous concentrer sur la manière d'en calculer une solution. Celle-ci, suivant la complexité des équations considérées, peut-être obtenue de manière théorique ou numérique [208]. À la vue de la difficulté des problèmes traités dans ce document, nous allons nous concentrer sur le second cas. Nous devons donc choisir une méthode numérique nous permettant de résoudre de façon approchée, dans un temps que nous considérons raisonnable, le modèle (par exemple, si celui-ci est composé d'équations différentielles partielles nous pouvons choisir entre divers schémas de discrétisation à base de différences, de volumes ou d'éléments finis [100, 202]). Lorsque le modèle numérique est programmé, une phase primordiale va être sa validation. En particulier, si cela est possible, il est intéressant

de comparer les sorties données par celui-ci avec des données réelles provenant du produit existant [118, 272].

- **Optimisation:** En assumant la validité de notre modèle, nous pouvons passer à la dernière étape qui consiste à améliorer certaines caractéristiques de notre produit. Pour cela et si le problème s'y prête, une manière efficace consiste à passer par la voie de l'optimisation. Tout d'abord nous devons reformuler notre question initiale en problème d'optimisation. Celui-ci peut être composé de divers types de contraintes (inégalités, égalités, non-linéaires, etc. dues, par exemple, à des limitations techniques), d'espaces de recherche (borné, connexe, discret, etc.) et de fonctionnelles (convexe, différentiable, etc.). Une fois le problème d'optimisation défini nous devons choisir une méthode de résolution (telle que descente, génétique ou simplexe [102, 174]) adaptée à celui-ci. La solution donnée par notre méthode doit, lorsque cela est possible, ensuite être analysée (par exemple par une étude de sensibilité [243]) et traitée dans le but d'une possible implémentation pratique.

Le processus décrit très schématiquement ci-dessus me sert de point de départ dans le développement et la présentation de mes thèmes de recherche.

Afin de pouvoir apprécier l'évolution de mon cursus en tant que chercheur, il est d'abord intéressant de faire un bref rappel, dans la Section 1.2, sur mes travaux de thèse. Ensuite, les résultats obtenus après le doctorat comme continuation de ces thématiques sont présentés dans la Section 1.3. Finalement, la Section 1.4 introduit les nouveaux sujets de recherche étudiés depuis le début de mon étape postdoctorale.

Les thématiques décrites dans les Sections 1.3 et 1.4 sont par la suite détaillées dans les **Chapitres 2-7**. Le **Chapitre 8** propose un bref bilan général des activités de recherche présentées dans ce document.

1.2 Rappels sur les travaux de thèse

Cette thèse, dirigée à l'Université Montpellier 2 par le professeur Bijan Mohammadi et intitulée "**Optimisation globale semi-déterministe et applications industrielles**", est disponible en version complète par téléchargement sur le lien suivant:

<http://www.mat.ucm.es/~ivorra/these/These-Ivorra.pdf>

ou en exemplaire imprimé à la bibliothèque de Universitaire de Montpellier (côte: **TS 2006.MON-61**). Les principaux thèmes d'étude sont résumés ci-dessous:

- **Algorithme multicouches linéaires pour améliorer l'efficacité de certaines méthodes d'optimisation:**

L'un des apports principaux de ce travail est le développement d'une méthode originale permettant d'améliorer, en termes de rapidité et précision, un large choix d'algorithmes d'optimisation. L'idée est d'utiliser un algorithme multicouches basé sur la méthode de la sécante [202] afin d'optimiser le choix d'une condition initiale d'une méthode d'optimisation considérée (par exemple, choisir le point de départ

d'une méthode de descente [174]). En particulier, cette méthodologie a été adaptée afin d'améliorer l'efficacité d'algorithmes de la plus grande pente [13], de la boule pesante [14] et génétiques [102]. Un programme Matlab regroupant toutes les techniques d'optimisation introduites durant cette thèse a été développé et validé à l'aide de fonctions tests [91]. Il est à noter que celui-ci sera le point de départ pour la création de l'outil "Global Optimization Platform" introduit plus tard dans la Section 1.3.

Ces travaux ont donné lieu à la publication suivante:

- Benjamin Ivorra, Damien Isèbe, Bijan Mohammadi. *Optimisation globale à complexité réduite. Calcul des structures. 2: 241-246. Hermès Science - Lavoisier. 2005.*

Dans la seconde partie de la thèse, les algorithmes d'optimisation développés précédemment furent appliqués à trois problèmes particuliers d'optimisation planté par diverses industries:

- **Synthèse de filtres optiques à base de fibres à réseaux de Bragg: (*collaboration avec l'entreprise Alcatel et l'Université de Paris 6*)**

Cette première application est basée sur le problème inverse suivant: étant donné le spectre réfléchi d'une fibre optique, dite à réseaux de Bragg, nous voulons reconstruire sa variation interne de l'indice de réfraction [96, 211]. Ce type de fibre est utilisé depuis de nombreuses années en télécommunication pour le transport de données et possède de nombreux avantages [80, 269]: un faible coût de fabrication, une simplicité de réalisation et un vaste domaine d'applications (tel que filtres passe bande, codage de signaux, etc.).

Dans ce contexte, nous avons introduit, puis résolu, trois problèmes d'optimisation liés à la conception de filtres optiques à l'aide des algorithmes d'optimisation développés pendant la thèse. Les résultats obtenus ont ensuite été comparés avec ceux donnés par un algorithme génétique et par une méthode industrielle classique [127]. Les solutions générées par notre algorithme ont montré un gain dans la précision du spectre réfléchi de l'ordre de 45%.

Ces résultats ont été publiés dans les articles:

- Benjamin Ivorra, Bijan Mohammadi, Patrick Redont, Laurent Dumas, Olivier Durand. *Semi-Deterministic vs. Genetic Algorithms for Global Optimization of Multichannel Optical Filters. International Journal of Computational Science and Engineering. 2(3): 170-178. Inderscience publishers. 2006.*
- Benjamin Ivorra, Patrick Redont, Laurent Dumas, Olivier Durand. *Semi-Deterministic Recursive Optimization Methods for Multichannel Optical Filters. Numerical Mathematics and Advanced applications . 2: 1007-1014. Springer Science. 2006.*
- Benjamin Ivorra, Angel M. Ramos, Bijan Mohammadi. *Design of Code Division Multiple Access Filters Using Global Optimization Techniques. Optimization en Engineering. Soumis.*

Cette étude a été financée par le projet CNRS:

- **80/0237:** *"Synthèse de guide optique de type WDM au moyen de deux méthodes d'optimisation différentes: génétique et semi-déterministe"*.
- **Optimisation de forme d'un mélangeur microfluidique utilisé pour le repliement de protéines: (collaboration avec l'Université de Stanford)**

Dans ce second problème, nous nous sommes intéressés à l'optimisation de forme d'un mélangeur microfluidique [181]. Ce dispositif est utilisé au niveau industriel ou scientifique pour lancer des processus de repliement de protéines (par exemple pour le séquençage d'ADN ou la fabrication de médicaments). Le repliement de protéines est actuellement un des principaux axes de recherche de la biologie moléculaire [39, 206, 58]. Les protéines sont des polymères de petite taille qui sont capables d'exister sous deux formes géométriques, l'une dépliée et biologiquement inactive, et l'autre repliée et biologiquement active. Elles possèdent la propriété de se replier lors d'un processus faisant intervenir, par exemple, certains solvants chimiques [220].

L'objectif de ce travail était de déterminer la forme d'un mélangeur microfluidique particulier, dit de Knight [157], pour laquelle un mélange, composé d'un dénaturant et d'une protéine, se dilue le plus rapidement possible dans un solvant, ceci afin de faciliter le processus de repliement d'une protéine donnée. Dans sa meilleure version cet appareil présentait un temps de mélange de $8\mu s$, notre but consistait à diminuer cette valeur [119]. Pour cela nous avons formulé un problème d'optimisation et l'avons résolu à l'aide d'algorithmes d'optimisations développés antérieurement. Le mélangeur optimisé présentait un temps de mélange de $1\mu s$ (soit une réduction de 87%). Il est aussi intéressant de noter qu'un prototype de ce mélangeur optimisé a été réalisé par le département d'ingénierie mécanique de l'Université de Stanford, et que les résultats expérimentaux ont confirmé l'amélioration de performances calculée numériquement.

Les articles suivant ont été publiés suite à ces travaux:

- David E. Hertzog, Benjamin Ivorra, Bijan Mohammadi, Olgica Bakajin, and Juan G. Santiago. *Optimization of a Microfluidic Mixer for Studying Protein Folding Kinetics*. *Analytical chemistry*. 78(13): 4299-4306. ACS Publications. 2006.
- Benjamin Ivorra, David E. Hertzog, Bijan Mohammadi, Juan G. Santiago. *Semi-deterministic and genetic algorithms for global optimization of microfluidic protein-folding devices*. *International Journal for Numerical Methods in Engineering*. 66(2): 319-333. Wiley InterScience. 2006.
- **Optimisation sous contraintes d'un portefeuille de crédits: (collaboration avec la Banque de Financement et d'Investissement de BNP-Paribas)**

En 2005, lors de la réalisation de cette thèse, le secteur financier européen connaissait une forte croissance [35]. La conception d'outils permettant de gérer des investissements de plus en plus importants et diversifiés était une nécessité afin de soutenir ce développement.

Dans ce contexte, nous nous sommes penchés sur l'optimisation, du point de vue des mesures de risque [11, 12, 239] de pertes (Value at Risk et Coherent Value at Risk) et rendements (profit et Risk Adjusted Return On Capital - RAROC), d'un portefeuille de crédits appelé Collateralized Debt Obligation (CDO). Plus précisément, nous avons formulé divers problèmes d'optimisation avec contraintes (concevoir un CDO afin de: réduire une mesure de risque en conservant le profit; maximiser le profit en maintenant la mesure de risque à un certain seuil; maximiser le RAROC en contrôlant la mesure de risque; maximiser le RAROC et le profit et minimiser la mesure de risque) et les avons résolu à l'aide d'une méthode d'optimisation étudiée pendant la thèse. Il est important de souligner que ces travaux ont été réalisés lors d'une collaboration avec l'équipe Portfolio Management de la Banque de Financement et d'Investissement du groupe BNP-Paribas, sous la forme d'un stage de valorisation de thèse de 5 mois. Les résultats obtenus durant cette étude ont permis d'atteindre des gains de performances de l'ordre de 20% à 40%, suivant le problème considéré, en comparaison des CDOs originaux conçus par des professionnels de BNP-Paribas.

L'article suivant portant sur cette étude a été publié:

- Benjamin Ivorra, Bijan Mohammadi and Angel M. Ramos. *Optimization strategies in credit portfolio management. Journal of Global Optimization. 43(2): 415-427. Springer Science. 2009.*

De plus, durant cette thèse, deux thématiques secondaires de recherche ont été étudiées:

- **Autres activités de recherche réalisées en parallèle de la thèse:**

Présenté dans une Annexe de la thèse, un problème de contrôle par point de l'équation visqueuse de Burgers [231] a été résolu à l'aide de divers algorithmes d'optimisation. Cette étude était centrée sur la nécessité d'utiliser (ou non) une méthode d'optimisation globale selon la valeur coefficient de pondération multipliant le terme de contrôle de la fonction à minimiser. Ce problème a été proposé par le professeur Angel Manuel Ramos del Olmo à l'occasion de mon stage de fin de Diplôme d'Études Approfondies (DEA) effectué en 2003 à l'Université Complutense de Madrid. Ce travail a été publié dans:

- Benjamin Ivorra, Angel M. Ramos and Bijan Mohammadi. *Semideterministic Global Optimization. Application to a Control Problem of the Burgers Equation. Journal of Optimization Theory and Applications. 135(3): 549-561. Springer Science. 2007*

Finalement, un autre problème portant sur le contrôle de température et d'émission de polluants à l'intérieur d'une flamme de bec bunsen [268] a été étudié à l'aide d'un algorithme d'optimisation introduit dans ma thèse. Les résultats obtenus (non reportés dans mon mémoire de doctorat) sont le fruit de la collaboration avec un autre étudiant en thèse dirigé par le professeur Bijan Mohammadi, Larvi Débianne, et ont été publiés dans l'article suivant:

- Larvi Debiante, Benjamin Ivorra, Bijan Mohammadi, Frank Nicoud, Thierry Poin-sot, Alexandre Ern, Hernst Pitsch. *A low-complexity global optimization algorithm for temperature and pollution control in flames with complex chemistry. International Journal of Computational Fluid Dynamics. 20(2): 93-98. Taylor & Francis. 2006.*

1.3 Approfondissement des thèmes étudiés en thèse

Nous allons maintenant introduire les travaux et résultats obtenus après la thèse (soutenue en Juin 2006) et qui sont une continuation des thématiques introduites dans la Section 1.2.

- **Développement du programme "Global Optimization Platform":** (*collaboration avec l'Université de Montpellier 2*)

Depuis la fin de mon doctorat, une de mes activités régulières est le suivi, le développement et la diffusion du programme d'optimisation développé pendant celui-ci. Ce programme, renommé "**Global Optimization Platform (GOP)**", se présente sous la forme d'une boîte d'outil Matlab sous licence "GNU General Public License V3.0" et est disponible gratuitement sur la page web du groupe de recherche MOMAT de l'Université Complutense au lien:

<http://www.mat.ucm.es/momat/software.htm>

De nouvelles méthodes d'optimisation (Pattern Search [15], Controlled Random Search [225] et Simulated Annealing [246]) ont été ajoutées dans GOP en supplément de celles déjà incluses lors de la thèse (algorithmes de la plus grande pente [174], de la boule pesante [14], ou génétiques [102]). De plus, la structure principale du programme a été repensée pour pouvoir fonctionner en parallèle sur des serveurs ou machines à CPU multi-coeurs [103] .

Une partie de ces travaux, présentés dans le **Chapitre 2** de ce mémoire, a donné lieu à la rédaction de l'article suivant (actuellement en version prépublication et devant être soumis sous peu):

- Benjamin Ivorra, Bijan Mohammadi, Angel M. Ramos. *Optimizing Initial Guesses to Improve Global Minimization. Prépublications du Departamento de Matemática Aplicada - UCM- N° MA-UCM-2008-06. 2008.*

- **Optimisation de forme par paramétrisation géométrique:**

La méthodologie appliquée pendant la thèse pour optimiser la forme d'un mélangeur microfluidique [135, 137, 201] (et aussi dans la littérature sur des problèmes divers, voir par exemple [16, 53, 70]) a été réutilisée afin d'étudier quatre cas de conception industrielle (deux cas repris de la thèse et deux autres nouveaux). Ces travaux sont détaillés dans le **Chapitre 3**.

Mélangeur microfluidique: (*collaboration avec l'Université de Stanford et l'Université d'Almeria*)

Le problème de conception d'un mélangeur microfluidique [118, 137] a été repris et enrichi par l'ajout de nouveaux paramètres géométriques et la prise en compte des vitesses d'injections [275]. Cela nous a permis de réduire le temps de mélange à $0.1 \mu s$. De plus, lors de cette étude un modèle complexe en 3 dimensions (alors que le modèle de base était une approximation en 2 dimensions) a été développé dans le but d'analyser la sensibilité numérique de notre appareil. Cette dernière étude a montré la robustesse de notre mélangeur et une validation expérimentale devrait bientôt compléter et confirmer nos résultats.

Les articles suivant concernent ce travail:

- Benjamin Ivorra, Juana Redondo, Juan Santiago, Pilar Ortigosa, and Angel Ramos. *Two- and three-dimensional modeling and optimization applied to the design of a fast hydrodynamic focusing microfluidic mixer for protein folding. Physics of Fluids. Soumis.*
- Juan Bello Rivas, Juan-Antonio Infante, Benjamin Ivorra, Juana Lopez Redondo, Pilar Martinez Ortigosa, Angel M. Ramos, Jose Maria Rey, Nadia Smith. *Mathematical modeling for protein folding devices. Applications to high pressure processing and microfluidic mixers. CD-ROM: Proceedings of the Congress EngOpt 2010. ISBN: 9789899626430. 2010.*

Structures côtières: (*collaboration avec l'entreprise BRL engineering et l'Université de Montpellier 2*)

Ce problème, développé durant la thèse doctorale de Damien Isèbe dirigée par le professeur Bijan Mohammadi [131], porte sur l'optimisation de forme de structures sous-marines dont le but est de limiter l'érosion côtière [223, 230]. Cette étude se place dans le contexte particulier des plages du littoral Sétouais dans l'Hérault et a été financée par l'ANR suivante:

- **ANR COPTER NT05-2-42253:** *"Conception, optimisation et prototypage d'ouvrage de lutte contre l'érosion en domaine littoral."*

Les structures optimisées ont permis de réduire de près de 21% l'impact de vagues dites destructrices sur le littoral en comparaison de la situation initiale. Ces résultats numériques ont été validés expérimentalement avec succès (reporté dans la thèse de Damien Isèbe).

Ce travail est décrit dans les articles suivants:

- Damien Isèbe, Pascal Azerad, Frédéric Bouchette, Benjamin Ivorra, Bijan Mohammadi. *Shape optimization of geotextile tubes for sandy beach protection. International Journal for Numerical Methods in Engineering. 74(8): 1262-1277. Wiley InterScience. 2008.*

- Damien Isèbe, Benjamin Ivorra, Pascal Azerad, Bijan Mohammadi and Frédéric Bouchette. *Progress in Global Optimization and Shape Design. Modeling, Simulation and Optimization of Complex Processes.* 303-312. Springer Berlin Heidelberg. 2008.

Bioréacteurs dispersifs: (collaboration avec l'INRA Montpellier et l'Université du Chili)

Cette étude porte sur la conception optimale d'un réacteur biologique, appelé bioréacteur, dans lequel un contaminant est dégradé par un écosystème microbien dans un environnement non homogène [121, 175, 209]. Pour simplifier, le bioréacteur est composé d'un silo rempli d'un liquide contenant les agents microbiens, de tuyaux d'entrée où est injectée une eau souillée par le contaminant et d'un tuyau de sortie où l'on récupère un liquide en parti dépollué. L'objectif est ici d'optimiser la géométrie et les vitesses d'injection du bioréacteur dans le but d'améliorer certaines de ses caractéristiques. Pour le moment nous avons considéré le problème de concevoir un bioréacteur en réduisant son volume initial (et ainsi son coût de fabrication) tout en maintenant sa capacité de dégradation du contaminant. Nous avons obtenu une baisse de 20% du volume.

Ces résultats préliminaires ont été publiés dans:

- Juan Bello Rivas, Jérôme Harmand, Benjamin Ivorra, Angel M. Ramos, Alain Rapaport. *Bioreactor shape optimization Modeling, simulation, and shape optimization of a simple bioreactor for water treatment. Les STIC pour l'environnement 2011.* 125-141. Transvalor - Presses des MINES. 2011. ISBN: 9788497494205

Filtres "Code Division Multiple Access":

Nous considérons le cas de la conception d'une fibre à réseau de Bragg (FRB) [81, 277] dont le spectre de sortie représente un Code "Division Multiple Access" (CDMA) [150, 161]. Notre but étant d'optimiser le profil (c'est à dire la forme) de son indice de réfraction afin d'obtenir le spectre désiré [241]. Ce sujet, proposé pendant mon doctorat [134], n'a pas été approfondi jusqu'à mon étape postdoctorale. Les résultats obtenus ont permis de générer des spectres proches de ceux désirés. À la différence des FRBs étudiées précédemment dans ma thèse, il n'existe pas de comparaison possible avec des techniques industrielles existantes [46, 248]. Ceci étant dû au fait que l'utilisation d'une seule FRB pour représenter un CDMA complet n'est pas actuellement considérée dans l'industrie. Notre approche permettrait théoriquement (mais non validé expérimentalement) la réduction du coût et du temps de fabrication d'appareils utilisant un codage CDMA.

L'article suivant a été soumis:

- Benjamin Ivorra, Angel M. Ramos, Bijan Mohammadi. *Design of Code Division Multiple Access Filters Using Global Optimization Techniques. Optimization en Engineering. Soumis.*

1.4 Nouvelles thématiques post-doctorales

Nous présentons dans cette Section les travaux portant sur de nouveaux sujets de recherches étudiés après mon doctorat.

- **Modélisation en épidémiologie vétérinaire:** (*collaboration avec le groupe de recherche VISAVET (UCM) et le ministère espagnol "Ministerio de Agricultura, Alimentación y Medio Ambiente".*)

Ce sujet, introduit dans le **Chapitre 4**, a été financé au travers des projets nationaux espagnols suivants:

- **CONS-C6-0356:** *"Un modelo matemático híbrido para la difusión de enfermedades animales y su impacto económico."*
- **MTM2011-22658:** *"Matemáticas para el avance interdisciplinar en altas presiones, sanidad animal y otros temas de interés científico y tecnológico"*
- **MTM2008-04621:** *"Modelización matemática en tecnología agroalimentaria y sanidad animal."*

En premier lieu, nous nous sommes concentrés sur la création d'un modèle, appelé Be-FAST (Between-Farm-Animal Spatial Transmission), permettant d'étudier la diffusion et le contrôle de la peste porcine classique (PPC) [199] entre les élevages de porcs de la province Espagnole de Ségovie [187]. Ce cas concret est d'un grand intérêt sanitaire et financier pour l'Espagne (second pays producteur de porcs en Europe), étant donné l'importance économique de la production de porcs dans cette province et le fait que la PPC ait touché Ségovie par le passé [184]. Ce modèle a été développé en se basant sur des études similaires provenant d'autres pays Européens [61, 147] et sur une base de données (contenant diverses informations relatives aux fermes, transports d'animaux et épidémies antérieures) fournie par le ministère "Ministerio de Agricultura, Alimentación y Medio Ambiente" [143, 182]. Les principales nouveautés introduites au niveau modélisation par rapport aux travaux déjà existant ont été: la considération de la diffusion de la PPC aussi bien entre les exploitations agricoles qu'à l'intérieur de chacune d'elles; l'utilisation de la base de données pour créer un réseau commercial réaliste; et la calibration des coefficients du modèle au contexte agricole Espagnol. Une fois Be-FAST programmé en Matlab, il a été validé en comparant les résultats obtenus lors de diverses expériences numériques à ceux produits à l'aide d'autres méthodes (dont, InterSpread Plus [244, 188], l'un des logiciels épidémiologique payants le plus utilisé au niveau mondial) et à diverses données d'épidémies de PPC ayant eu lieu par le passé en Europe. Notre modèle s'est ajusté le mieux à ces informations réelles. De plus, une étude complète de la sensibilité des paramètres et des sorties de Be-FAST ont permis de montrer la robustesse de celui-ci [243].

Ces résultats ont donné lieu aux publications suivantes:

- *Beatriz Martinez-Lopez, Benjamin Ivorra, Diène Ngom, Angel M. Ramos, Jose M. Sanchez-Vizcaino. A novel spatial and stochastic model to evaluate the within and*

between farm transmission of classical swine fever virus: II Validation of the model. Veterinary Microbiology. 155: 21-32. Elsevier. 2012.

- *Beatriz Martinez-Lopez, Benjamin Ivorra, Angel M. Ramos, Jose M. Sanchez-Vizcaino. A novel spatial and stochastic model to evaluate the within and between farm transmission of classical swine fever virus: I. General concepts and description of the model. Veterinary Microbiology. 147: 300-309. Elsevier. 2011.*
- *Benjamin Ivorra, Beatriz Martinez-Lopez, Jose M. Sanchez-Vizcaino, Angel M. Ramos. Mathematical formulation and validation of the Be-FAST model for Classical Swine Fever Virus spread between and within farms. Annals of Operations Research. Soumis.*

Nous sommes actuellement en train d'incorporer à Be-FAST l'impact économique d'une épidémie de PPC sur une région [125] (non reporté dans ce manuscrit). L'objectif étant de pouvoir optimiser le processus de contrôle de cette maladie par les autorités. Certains résultats préliminaires sont présentés dans la prépublication suivante:

- *E. Fernández Carrión, B. Ivorra, A. M. Ramos, B. Martínez-López, Sánchez-Vizcaino. Diseño de un modelo económico y de planes de control para una epidemia de PPC. Prépublications du Departamento de Matemática Aplicada - UCM - N° MA-UCM-2011-15. 2011.*

Un autre modèle épidémiologique est aussi en cours d'étude (non présenté dans ce document). Il simule l'évolution en Espagne de la population d'un moustique typiquement Africain pouvant être le vecteur de diverses maladies animales et/ou humaines telle que la langue bleue [33]. Du fait des changements climatiques actuels sur la péninsule Ibérique, ce moustique représente un réel risque sanitaire [97]. Ce modèle, basé sur le couplage de diverses EDPs [100] (permettant de modéliser le transport, la diffusion et la survie du moustique), prend en considération des données réelles portant sur la température, le vent et l'humidité des zones géographiques étudiées.

Des résultats préliminaires ont été publiés:

- *Cianci C., Granero B.R., Picado A.R., Pino Carrasco F.J., Rodrigo C.N., Tamayo M.E., Vázquez M., Ivorra B., Martínez-López B., Ramos A.M. and Sánchez-Vizcaino Rodriguez J.M. Impact of the climatic change on animal diseases spread: the Example of bluetongue in Spain. Revista Complutense de Ciencias Veterinarias. 5(1): 120-131. Revistas Científicas Complutenses. 2011.*
- **Traitements alimentaires à base de Haute-Pressions/Températures:** (*collaboration avec le CSIC (Espagne), plusieurs groupes de recherche de l'UCM et de l'Université de Puebla au Mexique, les entreprises Espagnoles Espuña et Hyperbaric, le gouvernement régional de Madrid et le ministère "Ministerio de Ciencia e Innovación"*).

Cette étude, décrite dans le **Chapitre 5**, est financée par les projets:

- **S2009/PPQ-1551:** *"Química a alta presión."*
- **MTM2011-22658:** *"Matemáticas para el avance interdisciplinar en altas presiones, sanidad animal y otros temas de interés científico y tecnológico"*
- **MTM2008-04621:** *"Modelización matemática en tecnología agroalimentaria y sanidad animal."*
- **PROMEP/103.5/09/1265:** *"Problemas directos e inversos en Biología e Ingeniería."*
- **MTM2007-64540:** *"Estudio matemático de problemas planteados en Ingeniería de alimentos."*
- **AGL2006-12112-C03-02/ALI:** *"Modelización matemática de procesos de congelación a altas presiones en la crioconzentración de zumos y la producción de helados."*

L'objectif principal est d'étudier l'impact sur des aliments de traitements faisant intervenir de brusques changements de pression et de température dans le but de les stériliser [64, 113]. Comme première étape de ce travail, nous avons considéré et modélisé un appareil industriel réel utilisé pour effectuer ce type de traitement alimentaire [108]. Le modèle, faisant intervenir diverses EDPs couplées, a été approximé numériquement à l'aide d'une méthode d'éléments finis. Les résultats numériques obtenus ont pu être validés en laboratoire [219].

La seconde étape a été l'incorporation au modèle précédent de l'étude de certaines activités biologiques à l'intérieur de l'aliment traité. En particulier, nous avons considéré des protéines intervenant lors de processus de péremption d'aliments [115, 128]. Il a été observé numériquement qu'en fonction de la protéine étudiée la température et la pression du traitement doivent être adaptées afin d'en améliorer l'efficacité.

Actuellement, je codirige avec Angel M. Ramos la **thèse de Juan M. Bello Rivas** (financée par le projet **S2009/PPQ-1551**) ayant pour but de définir un modèle simulant au niveau moléculaire [259] l'impact de hautes pressions et températures sur la structure des protéines (en particulier nous considérons le processus de repliement dont nous avons parlé auparavant).

Certains de ces travaux sont disponibles dans les articles:

- Juan A. Infante, Benjamin Ivorra, Angel M. Ramos, Jose M. Rey. *On the Modelling and Simulation of High Pressure Processes and Inactivation of Enzymes in Food Engineering. Mathematical Models and Methods in Applied Sciences.* 19(12): 2203-2229. World Scientific. 2009.
- Angel M. Ramos, Andres Fragueta, Juan A. Infante, Benjamin Ivorra, Jose M. Rey. *Inverse problems in High Pressure Processes and Food Engineering. First Symposium on Inverse Problems and Applications.* 39-56. Casa abierta al tiempo. ISBN: 786074775051. 2011.

- *Nadia Smith, Angel M. Ramos, Andres Fraguela, Juan A. Infante, Benjamin Ivorra, Jose M. Rey. Mathematical Modelling in Food Engineering. Numerical Simulation in Physics and Engineering. 283-301. Universidad de la Coruña. ISBN: 9788497494205. 2010.*
- *Juan M. Bello-Rivas, Benjamin Ivorra, Ángel M. Ramos, Antonio Rey. Estimation of the dynamics of protein folding processes from free energy computations. International Journal of Food Engineering. Soumis.*
- **Optimisation de trajectoire d'un bateau 'Skimmer' pour la dépollution en haute mer: (collaboration avec l'Université Nationale Autonome de Mexico, l'entreprise Mexicaine PeMEX et l'Université de Houston.)**

Ce travail, présenté dans le **Chapitre 6**, a été en parti financé par le projet:

- **Projet privé de l'entreprise PeMEX:** *"Generación de códigos de optimización para la caracterización de yacimiento naturalmente fracturados usando pruebas de variaciones de presión".*

Nous étudions ici le cas d'un bateau, dit 'Skimmer', dont le rôle est de pomper et nettoyer de l'eau souillée par un polluant (par exemple, du pétrole) en haute mer [9, 260]. Ce bateau dispose d'une autonomie de mouvement limité. Notre objectif est de définir une trajectoire optimale pour ce bateau, étant donné une zone géographique et un temps d'action connus, afin de nettoyer la plus grande quantité de polluant possible. Pour cela, nous avons d'abord reformulé ce problème à l'aide d'une modèle continu faisant intervenir une EDP de type advection-diffusion-réaction [40, 126, 265]. Ce modèle a ensuite été étudié numériquement à l'aide d'un schéma en volume fini de type upwind de premier ordre [100]. Puis, en se basant sur des cas de pollution réelle, nous avons défini trois problèmes particuliers d'optimisation et nous les avons résolu à l'aide d'un des algorithmes d'optimisation présenté dans le Chapitre 2.

Actuellement, nous sommes en train d'améliorer ce modèle en considérant un schéma numérique Warming and Beam avec limiteurs et une technique de splitting (non reporté dans ce document).

Les articles suivant traitent de ce problème:

- *Susana Gomez, Benjamin Ivorra, Angel M. Ramos. Optimization of a pumping ship trajectory to clean oil contamination in the open sea. Mathematical and computer modelling. 54(1): 477-489. Elsevier. 2011.*
- *Chitra Alavani, Roland Glowinski, Susana Gomez, Benjamin Ivorra, Pallavi Joshi, Angel M. Ramos. Modelling and Simulation of a Polluted Water Pumping Process. Mathematical and computer modelling. 51: 461-472. Elsevier. 2010.*
- **Optimisation de structures soumises à des charges aléatoires: (collaboration avec l'Université du Chili et l'Université des Andes de Santiago du Chili.)**

Nous considérons dans le **Chapitre 7** le cas d'une structure, aussi bien composée d'une armature [22] que d'un ensemble continu de matière [20, 247], de poids fixe et étant soumise à certaines charges de comportement aléatoire (suivant des lois normales). Notre objectif étant de trouver la meilleure distribution de poids possible afin que cette structure soit le plus robuste possible. Avant de résoudre ce problème, il a fallu définir un modèle permettant de mesurer ce que nous appelons la 'robustesse' d'une structure [19]. Nous avons pour cela commencé par étudier le cas d'armatures avant de passer au cas de structures continues (pouvant être vues comme des armatures avec une infinité de barres). Les deux approches sont similaires et font intervenir des EDPS d'élasticité pouvant être résolues numériquement à l'aide de schémas en différence finies ou en éléments finis suivant le type de structure [142, 247]. Une fois la solution numérique obtenue, nous avons défini différentes mesures de robustesse, faisant intervenir la déformation physique de la structure. Étant donné le caractère stochastique des charges considérées, ces mesures sont des combinaisons linéaires de l'espérance et de la variance de la déformation. Ensuite, nous avons étudié l'intérêt de considérer une bonne pondération entre ces deux valeurs dans le but de créer une structure qui résiste à la fois à la charge principale et à ses possibles perturbations. Le cas d'armatures a été traité entièrement en 2D et 3D, alors que le cas continu est toujours en cours d'étude.

Ces travaux sont présentés dans les articles:

- Carrasco Miguel, Benjamin Ivorra, Angel M. Ramos. *Variance-Expected Compliance Model for Structural Optimization. Journal of Optimization Theory and Applications. 152: 136-151. Springer Science. 2012.*
- Carrasco Miguel, Benjamin Ivorra, Rodrigo Lecaros, Angel M. Ramos. *An expected compliance model based on topology optimization for designing structures submitted to random loads. Differential Equations & Applications.4(1): 111-120. Ele-math. 2012.*

Ils ont été financés grâce au projet:

- **FONDECYT (CHILE) N°11090328:** *Numerical study of prox type algorithms and structural engineering applications.*

Chapter 2

Optimization Methods

Abstract - *Here, we introduce the global optimization algorithms used to solve the optimization problems presented in the next Chapters of this report. First, we envision global optimization as finding a suitable initial guess of a considered optimization algorithm. One can imagine that this possibility improve the capacity of existing optimization algorithms, including stochastic ones. Then, we present a novel methodology that consists in improving the initial guess of a particular optimization method by using a multi-layers semi deterministic algorithm based on line search methods. Finally, this approach is validated on several benchmark problems and the results are compared with those obtained by various metaheuristic algorithms.*

2.1 Introduction

Many optimization algorithms, such as the steepest descent and the Newton algorithms [52, 174], can be viewed as discrete forms of Cauchy problems for a system of Ordinary Differential Equations in the space of the control parameters [13, 37]. In situations where several local minima exist, those algorithms perform better if their initial condition belongs to the attraction basin of the infimum. This particular case shows the importance of the initial guesses in optimization algorithms. Hence, we would like to see if it is possible to provide a suitable initial condition in order to improve the efficiency of existing optimization methods.

This question has already been studied by considering various techniques. For instance, some methods, such as Direct Tabu Search (**DTS**) [98, 99, 116] and Tunelling [4, 169] algorithms, are based on the modification of the function to minimize, by adding penalty terms, in order to avoid the considered local search algorithm to enter in regions of the search space that have already been visited. Other techniques, like Greedy Randomized Adaptive Search Procedure (**GRASP**) [84, 85, 122] or the Universal Evolutionary Global Optimizer (**UEGO**) [233, 234] algorithms are based on the construction of a greedy solution combined with a local search phase which starts at the constructed solution. All those methods are generally used to improve the efficiency of local search methods.

In this Chapter, we discuss a novel technique based on the reformulation of the resolution of the optimization problem as a sub-optimization problem that consists in finding a suitable initial condition for the considered optimization algorithm. This new problem

is solved considering an original multi-layers semi deterministic algorithm based on line search methods. In particular, we focus on the implementation and the validation of our approach by considering three classes of deterministic and nondeterministic minimization algorithms: steepest descent [52, 174], heavy ball [13] and genetic algorithms (**GA**) [78, 88, 102, 183].

In Section 2.2, we give the general description of our method. Section 2.3 presents different ways to implement this idea. Finally in Section 2.4, the algorithms introduced previously are validated on various benchmark test cases [90, 91] and compared with other metaheuristic algorithms: DTS, Continuous GRASP (**C-GRASP**) and GA algorithms.

2.2 General optimization method

We introduce the following minimization problem:

$$\min_{x \in \Omega} h_0(x) \quad (2.2.1)$$

where $h_0 : \Omega \rightarrow \mathbb{R}$ is the cost function (not necessarily continuous) and x is the optimization parameter belonging to an admissible space $\Omega \subset \mathbb{R}^N$, with $N \in \mathbb{N}$.

We consider an optimization algorithm $A_0 : V \rightarrow \Omega$, called '*core optimization algorithm*', to solve (2.2.1). Here V is the space where we can choose the initial condition for A_0 (various examples are given in Section 2.3). The other optimization parameters of A_0 (such as the stopping criterion, number of iteration, etc.) are fixed by the user.

We assume the existence of a suitable initial condition $v \in V$ such that, for a given precision $\epsilon > 0$, $|A_0(v) - \min_{x \in \Omega} h_0(x)| < \epsilon$. In this case, solving numerically (2.2.1) with the considered core optimization algorithm means to solve

$$\begin{cases} \text{Find } v \in V \text{ such that} \\ A_0(v) \in \operatorname{argmin}_{x \in \Omega} h_0(x). \end{cases} \quad (2.2.2)$$

In order to solve (2.2.2), we propose to use a multi-layers semi-deterministic algorithm based on line search methods (see, for instance, [202]) called, for the sake of simplicity, '*Multi-Layer Secant Algorithm*' (**MSA**).

2.2.1 General description of the MSA

We introduce $h_1 : V \rightarrow \mathbb{R}$ by

$$h_1(v) = h_0(A_0(v)). \quad (2.2.3)$$

Thus problem (2.2.2) can be rewritten as:

$$\begin{cases} \text{Find } v \in V \text{ such that} \\ v \in \operatorname{argmin}_{w \in V} h_1(w). \end{cases} \quad (2.2.4)$$

A geometrical representation of $h_1(\cdot)$ in one dimension is shown in Figure 2.1 for a situation where the core optimization algorithm is the steepest descent algorithm [52, 174]

applied with a large number of iterations. We see that $h_1(\cdot)$ is discontinuous with plateaus. Indeed, the same point is reached by the algorithm starting from any of the points of the same attraction basin. Furthermore, $h_1(\cdot)$ is discontinuous where the functional reaches a local maximum. One way to minimize such a kind of function, in the one dimensional case, is to consider line search optimization methods (such as secant method or dichotomy [202]).

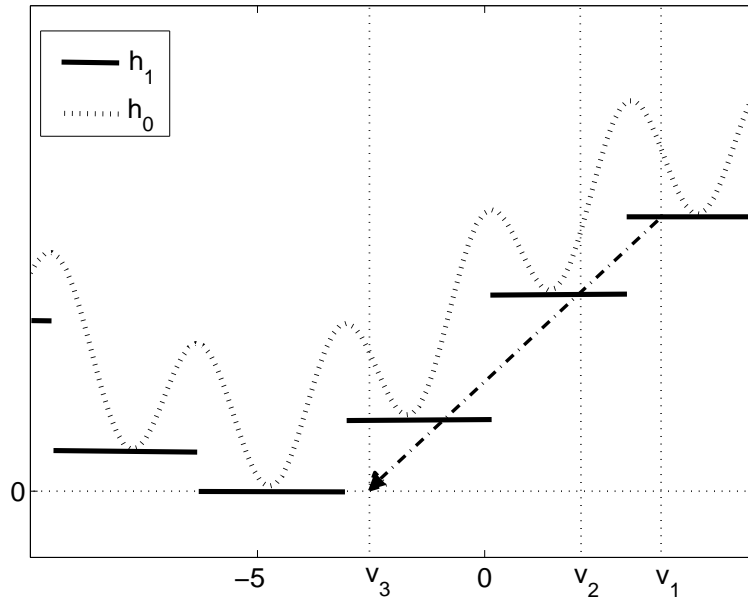


Figure 2.1: (....) $h_0(x) = \frac{1}{2} \cos(2x) + \sin(\frac{1}{3}x) + 1.57$ for $x \in \Omega = V = [-10, 6]$. (—) Geometrical representation of $h_1(\cdot)$ when the steepest descent method is used as core optimization algorithm with 10000 iterations. (-.-) Geometrical representation of one execution of the algorithm $A_1(v_1)$, written in Section 2.3.1, when v_1 is given and $t_{l_1} = 1$. v_2 is generated randomly in $[-10, 6]$ during the first Step of the algorithm. v_3 is built by the secant method performed during the Step 2.2. During the Step 3, since $h_1(v_3)$ is lower than $h_1(v_1)$ and $h_1(v_2)$, v_3 is considered as the best initial condition and it is returned as the output.

Thus, in order to solve (2.2.4), we introduce the algorithm $A_1 : V \rightarrow V$ that, for each $v_1 \in V$, returns $A_1(v_1)$ given by

Step 1- Choose v_2 randomly in V .

Step 2- Find $v \in \operatorname{argmin}_{w \in \mathcal{O}(v_1, v_2)} h_1(w)$, where $\mathcal{O}(v_1, v_2) = \{v_1 + t(v_2 - v_1), t \in \mathbb{R}\} \cap V$, using a line search method.

Step 3- Return v .

The line search minimization algorithm in A_1 and its corresponding parameters are defined by the user.

In fact, to be more efficient, we are interested to perform a multi-directional search of the solution of (2.2.2). To do so, we add a layer external to the algorithm A_1 by considering the following methodology:

We define $h_2 : V \rightarrow \mathbb{R}$ by

$$h_2(v) = h_1(A_1(v)) \quad (2.2.5)$$

Then we consider the problem:

$$\begin{cases} \text{Find } v \in V \text{ such that} \\ v \in \operatorname{argmin}_{w \in V} h_2(w). \end{cases} \quad (2.2.6)$$

To solve (2.2.6) we use the two-layers algorithm $A_2 : V \rightarrow V$ that, for each $v_1 \in V$, returns $A_2(v_1)$ given by

Step 1- Choose v_2 randomly in V .

Step 2- Find $v \in \operatorname{argmin}_{w \in \mathcal{O}(v_1, v_2)} h_2(w)$ using a line search method.

Step 3- Return v .

As previously, the line search minimization algorithm in A_2 and its corresponding parameters are defined by the user. Due to the fact that the line search direction $\mathcal{O}(v_1, v_2)$ in A_1 is constructed randomly, the algorithm A_2 perform a multi-directional search of the solution of (2.2.2).

This construction can be pursued recursively by defining for $i = 3, 4, \dots$

$$h_i(v) = h_{i-1}(A_{i-1}(v)) \quad (2.2.7)$$

and considering the problem:

$$\begin{cases} \text{Find } v \in V \text{ such that} \\ v \in \operatorname{argmin}_{w \in V} h_i(w). \end{cases} \quad (2.2.8)$$

Problem (2.2.8) is solved using the i -layers algorithm $A_i : V \rightarrow V$ that, for each $v_1 \in V$, returns $A_i(v_1)$ given by

Step 1- Choose v_2 randomly in V .

Step 2- Find $v \in \operatorname{argmin}_{w \in \mathcal{O}(v_1, v_2)} h_i(w)$ using a line search method.

Step 3- Return v .

As before, the line search method used in Step 2 and its corresponding parameters are defined by the user.

In practice, as specified in Section 2.4, we run A_i with a suitable stopping criterion and with $v_1 \in V$ arbitrary (or $v_1 \in V$ a good initial guess, if available).

The choice of the random technique used to generate v_2 during Step 1 of A_i is important and could depend on the shape of h_0 . For instance, if we know that h_0 has several local minima in Ω with small attraction basins, it seems appropriate to generate v_2 in a small neighborhood of v_1 .

The line search minimization algorithm used during Step 2 of A_i could depend on the properties of h_0 , as discussed in Section 2.3.

2.3 MSA implementations

We present various MSA implementations, considering different core optimization algorithms, in the particular case where h_0 is a non negative function (or, equivalently, greater than a known real number), which often occurs in industrial problems (see, for instance, [138, 136, 137, 135]).

2.3.1 MSA implementation with first order descent core optimization algorithms

We consider core optimization algorithms A_0 that come from the discretization of the following initial value problem [13, 37, 201]:

$$\begin{cases} M(x(t), t) \frac{dx}{dt}(t) = -d(x(t)), & t \geq 0, \\ x(0) = x_0, \end{cases} \quad (2.3.1)$$

where t is a fictitious time, $M : \Omega \times \mathbb{R} \rightarrow M_{N \times N}$ (where $M_{N \times N}$ denotes the set of matrix $N \times N$) and $d : \Omega \rightarrow \mathbb{R}^N$ is a function giving a descent direction. For example, assuming $h_0 \in C^1(\Omega, \mathbb{R})$, if $d = \nabla h_0$ and $M(x, t) = Id$ (the identity operator) for all $(x, t) \in \Omega \times \mathbb{R}$ we recover, considering a suitable discretization, the steepest descent method [37].

According to previous notations, we use $V = \Omega$ and denote by $A_0(x_0) := A_0(x_0; t_0, \epsilon, \kappa)$ the solution returned by the core optimization algorithm starting from the initial point $x_0 \in \Omega$ after $t_0 \in \mathbb{N}$ iterations, considering a stopping criterion defined by $\epsilon \in \mathbb{R}$ and κ is the set of remaining algorithm parameters fixed by the user.

We point out that for this choice of A_0 , problem (2.2.2) is trivially admissible since, for instance, any $x_0 \in \operatorname{argmin}_{x \in \Omega} h_0(x)$ is a solution. In Section 2.3.2, we show a different case where the solvability of the corresponding problem is not trivial.

We consider a particular implementation of the algorithms A_i , $i = 1, 2, \dots$, introduced previously. For $i = 1, 2, \dots$, $A_i(v_1)$ is applied with a secant method in order to perform the line search. It reads:

Step 1- Choose $v_2 \in \Omega$ randomly.

Step 2- For l from 1 to $t_{l_i} \in \mathbb{N}$:

Step 2.1- If $h_i(v_l) = h_i(v_{l+1})$ go to **Step 3**

Step 2.2- Set $v_{l+2} = \operatorname{proj}_\Omega(v_{l+1} - h_i(v_{l+1}) \frac{v_{l+1} - v_l}{h_i(v_{l+1}) - h_i(v_l)})$

where $\operatorname{proj}_\Omega : \mathbb{R}^N \rightarrow \Omega$ is a projection algorithm over Ω defined by the user.

Step 3- Return the output: $\operatorname{argmin}\{h_i(v_m), m = 1, \dots, t_{l_i}\}$

In the previous algorithm, the value of t_{l_i} , for $i = 1, 2, \dots$, depends on the desired computational complexity. In Section 2.4, we check the efficiency of those algorithms for various set of values.

A geometrical representation of one execution of the algorithm A_1 in one dimension is shown in Figure 2.1.

From a theoretical point of view, as the secant method is adapted to find the zeros of a function [202], those algorithms perform better if the minimum value of h_0 , denoted by $h_0^* \in \mathbb{R}$, is known (for instance, in some inverse problems [135]). Indeed, we can assume that $h_0^* = 0$ (we minimize $h_0 - h_0^*$ instead of h_0) and thus A_i intends to find the zero of h_i (see [134] for more details).

In practice (see experiments presented in Section 2.4), if the only information available is that h_0 is greater than a known real number $h_l \in \mathbb{R}$, we consider $h_0 - h_l$ instead of h_0 and this algorithm still gives good numerical results. This efficiency can be in part explained by the structure of the secant iteration that gives quick informations on the behavior of h_0 : if there is a significant evolution of the cost function value between v_{l+1} and v_l , the secant iteration generates v_{l+2} close to v_{l+1} performing a fine search around v_{l+1} . If not, the secant method generates v_{l+2} far from v_{l+1} allowing to explore new regions.

Remark. Although this case is not considered in this Chapter, if no information about a lower bound value of h_0 is available we could consider other MSA implementations. For instance, we can replace the secant method used during Step 2.2 by the steepest descent iteration starting from v_{l+1} and using $-\frac{v_{l+1}-v_l}{h_i(v_{l+1})-h_i(v_l)}$ as the descent direction. This new step reads:

Step 2.2- Set $v_{l+2} = \text{proj}_\Omega(v_{l+1} - \rho \frac{v_{l+1}-v_l}{h_i(v_{l+1})-h_i(v_l)})$

where $\text{proj}_\Omega : \mathbb{R}^N \rightarrow \Omega$ is a projection algorithm over Ω defined by the user and $\rho \in \mathbb{R}$ is obtained solving numerically $\min_{\rho \in \mathbb{R}^+} h_0(\text{proj}_\Omega(v_{l+1} - \rho \frac{v_{l+1}-v_l}{h_i(v_{l+1})-h_i(v_l)}))$ by using, for example, a dichotomy method (see [202]).

2.3.2 MSA implementation with second order descent core optimization algorithms

In order to have an exploratory character during the optimization process, allowing to escape from attraction basins, we could use variants of the core optimization methods presented in Section 2.3.1 after adding a second order derivative term to the initial value problem (2.3.1). More precisely, we consider core optimization algorithms coming from the discretization of:

$$\begin{cases} \eta \frac{d^2 x}{dt^2}(t) + M(x(t), t) \frac{dx}{dt}(t) = -d(x(t)), & t \geq 0 \\ x(0) = x_0, & x_t(0) = x_{t,0}, \end{cases} \quad (2.3.2)$$

with $\eta \in \mathbb{R}$. For instance, assuming $h_0 \in C^2(\Omega, \mathbb{R})$, when $d = \nabla h_0$ and $M(x, t) = Id$ for all $(x, t) \in \Omega \times \mathbb{R}$ we recover the 'heavy ball' dynamical system and, considering an adequate discretization, we obtain the heavy ball optimization algorithm [14].

Here $A_0(x_0, x_{t,0}) := A_0(x_0, x_{t,0}; t_0, \epsilon, \kappa)$ denotes the solution returned by the core optimization algorithm starting from the initial point $x_0 \in \Omega$ with an initial velocity of $x_{t,0} \in \mathbb{R}^N$, after $t_0 \in \mathbb{N}$ iterations, considering a stopping criterion given by $\epsilon \in \mathbb{R}$ and κ is the set of remaining algorithm parameters fixed by the user.

In this case, we propose two different formulations for (2.2.2). The first one is given by

$$\begin{cases} \text{Find } x_0 \in \Omega \text{ such that} \\ A_0(x_0, x_{t,0}) \in \operatorname{argmin}_{w \in \Omega} h_0(w), \end{cases} \quad (2.3.3)$$

where $x_{t,0}$ is fixed. The second one is given by

$$\begin{cases} \text{Find } x_{t,0} \in \mathbb{R}^N \text{ such that} \\ A_0(x_0, x_{t,0}) \in \operatorname{argmin}_{w \in \Omega} h_0(w), \end{cases} \quad (2.3.4)$$

where x_0 is fixed.

The existence of $x_0 \in \Omega$ such that (2.3.3) admits a solution is trivial (as in Section 2.3.1). In the second case, under convenient hypotheses, it can be proved the existence of $x_{t,0} \in \mathbb{R}^N$ such that the solution of (2.3.4) can be numerically approached, as stated in the following theorem:

Theorem 1 *Let $h_0 \in C^2(\mathbb{R}^N, \mathbb{R})$ having a minimum, which is reached at $x_m \in \mathbb{R}^N$. We assume that ∇h_0 is Lipschitz continuous. Then, for every $(x_0, \epsilon) \in \mathbb{R}^N \times \mathbb{R}^+$, there exists $(\sigma, \tau_b) \in \mathbb{R}^N \times \mathbb{R}^+$ such that the solution of the following dynamical system:*

$$\begin{cases} \eta \frac{d^2 x}{dt^2}(t) + \frac{dx}{dt}(t) = -\nabla h_0(x(t)), & t \geq 0, \\ x(0) = x_0, \\ x_t(0) = \sigma, \end{cases} \quad (2.3.5)$$

with $\eta \in \mathbb{R}$, passes at time τ_b into the ball $B_\epsilon(x_m)$.

Proof : We assume $x_0 \neq x_m$ ($x_0 = x_m$ is a trivial case). Let $\delta \geq 0$, we consider the initial value problem:

$$\begin{cases} \eta \frac{d^2 y_\delta}{dt^2}(t) + \delta \frac{dy_\delta}{dt}(t) = -\delta^2 \nabla h_0(y_\delta(t)), & t \geq 0, \\ y_\delta(0) = x_0, \\ \frac{dy_\delta}{dt}(t)(0) = \varrho(x_m - x_0), \end{cases} \quad (2.3.6)$$

with $\varrho \in \mathbb{R}^+ \setminus \{0\}$. Let us show that y_δ passes at some time into the ball $B_\epsilon(x_m)$:

- If $\delta = 0$, we obtain the following system :

$$\begin{cases} \eta \frac{d^2 y_0}{dt^2}(t) = 0, & t \geq 0, \\ y_0(0) = x_0, \\ \frac{dy_0}{dt}(0) = \varrho(x_m - x_0). \end{cases} \quad (2.3.7)$$

System (2.3.7) describes a straight line of origin x_0 and passing at some time $\tau_\varrho \in \mathbb{R}^+$ by the point x_m , i.e. $y_0(\tau_\varrho) = x_m$.

- If $\delta \neq 0$, system (2.3.6) can be rewritten in the form

$$\frac{dw}{dt}(t) = \begin{pmatrix} \frac{dy_\delta}{dt}(t) \\ -\delta \frac{dy_\delta}{dt}(t) - \delta^2 \nabla h_0(y_\delta(t)) \end{pmatrix} = f(t, w(t), \delta), \quad (2.3.8)$$

with $w(t) = (y_\delta(t), \eta \frac{dy_\delta}{dt}(t))$ and f continuous in t and in δ and Lipschitz continuous in $w(t)$ [14]. Then, applying the Cauchy-Lipschitz theorem (see, for instance, [264]) and the continuity theorem 3.4 in [110]:

$$\lim_{\delta \rightarrow 0} |y_\delta(\tau_\varrho) - y_0(\tau_\varrho)| = 0.$$

Thus for every $\epsilon \in \mathbb{R}^+ \setminus \{0\}$, there exists $\delta_\epsilon \in \mathbb{R}^+$ such that for every $\delta \leq \delta_\epsilon$:

$$|y_\delta(\tau_\varrho) - x_m| < \epsilon. \quad (2.3.9)$$

Let $\epsilon \in \mathbb{R}^+ \setminus \{0\}$. We consider the change of variable given by $s = \delta_\epsilon t$ and $x(s) = y_{\delta_\epsilon}(\frac{s}{\delta_\epsilon})$. Then system (2.3.6) becomes:

$$\begin{cases} \eta \frac{d^2x}{ds^2}(s) + \frac{dx}{ds}(s) = -\nabla h_0(x(s)), & s \geq 0, \\ x(0) = x_0, \\ \frac{dx}{ds}(0) = \frac{\varrho}{\delta_\epsilon}(x_m - x_0). \end{cases} \quad (2.3.10)$$

Let $\tau_b = \delta_\epsilon \tau_\varrho$. Under this assumption, $x(\tau_b) = y_{\delta_\epsilon}(\tau_\varrho)$. Thus, due to (2.3.9): $|x(\tau_b) - x_m| < \epsilon$. We have found $\sigma = \frac{\varrho}{\delta_\epsilon}(x_m - x_0) \in \mathbb{R}^N$ and $\tau_b \in \mathbb{R}^+$ such that the solution of system (2.3.5) passes at time τ_b into the ball $B_\epsilon(x_m)$.

□

In order to determine a solution of (2.3.3) or (2.3.4), we can consider, for example, the same implementation of algorithms A_i (with $i = 1, 2, 3, \dots$) as the one introduced in Section 2.3.1.

2.3.3 MSA implementation with genetic core optimization algorithms

Now, we study an implementation of the MSA with a GA as core optimization algorithm.

Broadly speaking, GAs are search techniques which try to solve problems like (3.1.8) through a stochastic process based on an analogy with the Darwinian evolution of species [102]. The GAs have many advantages as for example: they do not require sensitivity

computation, they can solve complex optimization problems (e.g., with high dimensional search space or function with various with local minima), and they are intrinsically parallel. However, they also have some important drawbacks, as they exhibit slower convergence and lower accuracy than other method, such as gradient algorithms. Next, we describe the GA considered during this work:

- **Step 1- Inputs:** User must define four parameters: $N_p \in \mathbb{N}$, $N_g \in \mathbb{N}$, $p_m \in [0, 1]$ and $p_c \in [0, 1]$. The meaning of those parameters is clarified later in the following steps. In addition, a first set, called 'initial population' and denoted by $X^0 = \{x_j^0 \in \Phi, j = 1, \dots, N_p\}$, of N_p points (called 'individuals') in Φ is also provided by user.
- **Step 2- Generating new populations:** Starting from X^0 , we recursively create N_g new populations by applying four stochastic processes: 'selection', 'crossover', 'mutation' and 'elitism', which are described in Steps 3.1, 3.2, 3.3 and 3.4, respectively.

More precisely, let $X^i = \{x_j^i \in \Phi, j = 1, \dots, N_p\}$, with $i = 1, \dots, N_g - 1$, denotes the population at iteration i . Then, using the (N_p, N) -real valued matrix:

$$X^i = \begin{bmatrix} x_1^i \\ \vdots \\ x_{N_p}^i \end{bmatrix} = \begin{bmatrix} x_1^i(1) & \dots & x_1^i(N) \\ \vdots & \ddots & \vdots \\ x_{N_p}^i(1) & \dots & x_{N_p}^i(N) \end{bmatrix},$$

with $x_j^i = (x_j^i(1), \dots, x_j^i(N)) \in \Phi$, X^{i+1} is obtained by considering:

$$X^{i+1} = (I_N - \mathcal{E}^i)(\mathcal{C}^i \mathcal{S}^i X^i + \mathcal{M}^i) + \mathcal{E}^i X^i,$$

where matrices \mathcal{S}^i , \mathcal{C}^i , \mathcal{M}^i , \mathcal{E}^i and I_N are described as follows.

- **Step 2.1- Selection:** We randomly select N_p individuals from X^i with eventual repetitions. Each individual $x_j^i \in X^i$, with $j = 1, \dots, N_p$, has a probability to be selected during this process which is given by $J_{2D}^{-1}(x_j^i) / \sum_{k=1}^{N_p} J_{2D}^{-1}(x_k^i)$. This step can be summarized as

$$X^{i+1,1} = \mathcal{S}^i X^i,$$

where \mathcal{S}^i is a (N_p, N_p) -matrix with $\mathcal{S}_{j,k}^i = 1$ if the k -th individual of X^i is the j -th selected individual and $\mathcal{S}_{j,k}^i = 0$ otherwise.

- **Step 2.2- Crossover:** For each pair of consecutive individuals (rows) $2j - 1$ and $2j$ in $X^{i+1,1}$, with $1 \leq j \leq \text{floor}(N_p/2)$ (where $\text{floor}(X)$ is the nearest integer lower or equal than X), we determine, with a probability p_c , if those rows exchange data or if they are directly copied into an intermediate population denoted by $X^{i+1,2}$. Mathematically, this step can be written as:

$$X^{i+1,2} = \mathcal{C}^i X^{i+1,1},$$

where \mathcal{C}^i is a real-valued (N_p, N_p) -matrix. The coefficients of the $(2j - 1)$ -th and $2j$ -th rows of \mathcal{C}^i , with $1 \leq j \leq \text{floor}(N_p/2)$, are given by:

$$\mathcal{C}_{2j-1,2j-1}^i = \lambda_1, \quad \mathcal{C}_{2j-1,2j}^i = 1 - \lambda_1, \quad \mathcal{C}_{2j,2j}^i = \lambda_2, \quad \mathcal{C}_{2j,2j-1}^i = 1 - \lambda_2$$

where $\lambda_1 = \lambda_2 = 1$, with a probability $1 - p_c$, or λ_1 and λ_2 are randomly chosen in $]0, 1[$, considering a uniform distribution, in other case. Other coefficients of \mathcal{C}^i are set to 0. If N_p is odd then we also set $\mathcal{C}_{N_p, N_p}^i = 1$ and then the N_p -th row of $X^{i+1,1}$ is directly copied in $X^{i+1,2}$.

- **Step 2.3- Mutation:** We decide, with a probability p_m , if each row of $X^{i+1,2}$ is randomly perturbed or not. This step is defined by:

$$X^{i+1,3} = X^{i+1,2} + \mathcal{M}^i,$$

where \mathcal{M}^i is a real-valued (N_p, N) -matrix where the j -th row, $j = 1, \dots, N_p$, is equal to 0, with a probability $1 - p_m$, or a random vector $m_j \in \mathbb{R}^N$, generated considering a uniform distribution in the subset of \mathbb{R}^N such that $x_j^{i+1,2} + m_j \in \Phi$, otherwise.

- **Step 2.4- Elitism:** Let x_b^i , where $b \in 1, \dots, N_p$, be the individual in X^i with the lowest value of J_{2D} (or, if there exists various, one of those individuals selected randomly). If x_b^i has a lower J_{2D} value than all the individuals in $X^{i+1,3}$, it is directly copied at the b -th row of X^{i+1} . This step can be formalized as:

$$X^{i+1} = (I_N - \mathcal{E}^i)(X^{i+1,3}) + \mathcal{E}^i X^i,$$

where I_N is the identity matrix of size N and \mathcal{E}^i is a real-valued (N_p, N_p) -matrix such that $\mathcal{E}^i(b, b) = 1$ if x_b^i has a lower h_0 value than all the individuals in $X^{i+1,3}$ and 0 otherwise, $\mathcal{E}^i = 0$ elsewhere.

- **Step 3- Output:** After N_g iterations, the GA stops and returns an output solution denoted by

$$A_0(X^0, N_p, N_g, p_m, p_c) = \underset{i=1, \dots, N_g, j=1, \dots, N_p}{\operatorname{argmin}} \{h_0(x_j^i) / x_j^i \text{ is the } j\text{-th row of } X^i\}.$$

With the basic evolution processes described above, it is generally observed that the best obtained individual is getting closer after each generation to the optimal solution of the problem [102].

Those algorithms do not require sensitivity computation, perform global and multi-objective optimization and are easy to parallelize. However, their drawbacks remain their weak mathematical background, their computational complexity, their slow convergence and their lack of accuracy.

As a fine convergence is difficult to achieve with GA based algorithms, it is recommended when it is possible, to complete the GA iterations by a descent method. This is especially useful when the functional is flat around the infimum [70].

Considering the GA introduced previously, Problem (2.2.2) can be rewritten as:

$$\begin{cases} \text{Find } X^0 \in V = \Omega^{N_p} \text{ such that} \\ A_0(X^0) \in \operatorname{argmin}_{w \in \Omega} h_0(w) \end{cases} \quad (2.3.11)$$

where $A_0(X^0) := A_0(X^0, N_p, N_g, p_m, p_c)$.

The solution of (2.3.11) may be determined, for instance, by using the algorithm A_i (with $i = 1, 2, \dots$) presented in Section 2.3.1. However, a first numerical study (see [134] for more details) shows that the following variation of previous algorithm A_i (with $i = 0, 1, 2, 3, \dots$), denoted by B_i , is better suited to the GA case. Let $X_1^0 = \{x_{1,j}^0 \in \Omega, j = 1, \dots, N_p\}$ and $B_0(X^0) = A_0(X^0)$, then, for $i > 0$, $B_i(X_1^0)$ reads:

Step 1- For l from 1 to $t_{l_i} \in \mathbb{N}$:

Step 1.1- Set $o_l \in \operatorname{argmin}\{h_0(x) : x \in B_{i-1}(X_l^0)\}$.

Step 1.2- We construct $X_{l+1}^0 = \{x_{l+1,j}^0 \in \Omega, j = 1, \dots, N_p\}$ as following: $\forall j \in \{1, \dots, N_p\}$, if $h_0(o_l) = h_0(x_{l,j}^0)$ set $x_{l+1,j}^0 = x_{l,j}^0$
 else set $x_{l+1,j}^0 = \operatorname{proj}_\Omega(x_{l,j}^0 - h_0(o_l) \frac{o_l - x_{l,j}^0}{h_0(o_l) - h_0(x_{l,j}^0)})$
 where $\operatorname{proj}_\Omega : \mathbb{R}^N \rightarrow \Omega$ is a projection algorithm over Ω defined by the user.

Step 2- Return the output: $\operatorname{argmin}\{h_i(X_m^0), m = 1, \dots, t_{l_i}\}$

As previously, the value of t_{l_i} , for $i = 1, 2, \dots$, depends on the desired computational complexity.

This version of the algorithm intends to optimize, individual by individual, the initial population of B_{i-1} . For each individual in the initial population:

- If there is a significant evolution of the cost function value between this individual and the best element found by B_{i-1} , the secant method used in Step 1.2 generates, in the optimized initial population X_{l+1}^0 , a new individual close to this best element.
- If not, the secant method allows to create a new individual in X_{l+1}^0 far from the current solution given by B_{i-1} .

The numerical experiments in Section 2.4 seem to indicate that considering algorithms B_i , with $i > 0$, reduces the computational complexity of GAs. In particular, this allows to consider smaller values of N_p and N_g than with GAs alone (i.e. B_0).

2.4 Numerical validation of the method

In order to check the efficiency of the MSA implementations presented in Section 2.3, we apply them to the following set of benchmark optimization problems: Branin, Eason, Goldstein-Price, Shubert, Hartmann (3 and 6 variables), Rosenbrock (2, 5 and 10 variables), Shekel (4 variables and $m = 5, 7, 10$), and Zakharov (5 and 10 variables). A complete description of those problems can be easily found in the literature (see, for instance, [90, 91]). These problems represent a wide range of the main difficulties found in optimization problems [116]. Furthermore, they are frequently used to validate optimization algorithms [38, 116, 122, 246, 263]. In particular, with the purpose of comparing our algorithms with other efficient heuristic methods, we report results obtained with DTS [116] or C-GRASP [122].

2.4.1 Parameters in algorithms

We consider various versions of the MSA in the cases when the core optimization algorithm is the steepest descent algorithm, the heavy ball optimization algorithm and a genetic algorithm, respectively:

- When the steepest descent algorithm [174] is the core optimization algorithm we use the MSA implementation presented in Section 2.3.1. We consider the cases when the number of layers is $i = 1$ (the algorithm is then denoted by **SDMSA-1L**, as Steepest Descent MSA 1-Layer), $i = 2$ (**SDMSA-2L**) and $i = 3$ (**SDMSA-3L**). We set $t_0 = 10$ and $t_1 = 1000$ for SDMSA-1L; $t_0 = t_{l_1} = 10$ and $t_{l_2} = 1000$ for SDMSA-2L; and $t_0 = t_{l_1} = t_{l_2} = 10$ and $t_{l_3} = 1000$ for SDMSA-3L. The descent step size ρ is determined by using 10 iterations of a dichotomy method starting from $\rho_0 = 1$ [202].
- When the heavy ball optimization algorithm [14] is used as the core optimization algorithm, we use the two-layers algorithm A_2 , described in Sections 2.3.1 and 2.3.2, with $\eta = 0.1$, $t_0 = t_{l_1} = 10$ and $t_{l_2} = 1000$. The velocity $x_{t,0}$ is the initial condition to be optimized. This algorithm is denoted by **HBMSA** (Heavy-Ball MSA).
- With GA as the core optimization algorithm, we use the algorithm B_2 introduced in Section 2.3.3 with $t_{l_1} = 10$ and $t_{l_2} = 1000$. This algorithm is denoted by **GMSA** (Genetic MSA). In addition, the GA parameters are set as following:
 - The population size and generation number are set to $N_p = N_g = 10$.
 - The selection is a roulette wheel type [102] proportional to the rank of the individuals in the population.
 - The crossover is barycentric [70] in each coordinate with a probability of $p_c = 0.45$.
 - The mutation process is non-uniform [102] with a probability of $p_m = 0.35$.
 - A one-elitism principle [102], that consists in keeping the current best individual in the next generation, has also been imposed.

At the end of the algorithm, 10 iterations of the steepest descent algorithm, described in SDMSA-1L, are performed in order to improve the result.

The performances of the previous MSAs are compared with DTS and C-GRASP algorithm (whose implementation, parameters and results are presented in [116] and [122], respectively), and with a GA. This GA uses the same stochastic processes than GMSA but with a different sets of parameters: $N_{gen} = 1000$, $N_p = 180$, $p_c = 0.45$, $p_m = 0.15$ (other set of parameters have been previously tested, but for the sake of simplicity, we present here only the best one [134]). As in the GMSA case, at the end of the GA, 10 iterations of the steepest descent algorithm are carried out.

As the global minimum h_0^* of the different benchmark problems is known, the stopping criterion has been set to:

$$|h_0^* - \tilde{h}_0| \leq \epsilon_1 |h_0^*| + \epsilon_2, \quad (2.4.1)$$

where \tilde{h}_0 is the current solution of the algorithm, $\epsilon_1 = 10^{-4}$ and $\epsilon_2 = 10^{-6}$ [122]. However, in the case of the GMSA and GA, those algorithms are run with $\epsilon_1 = 10^{-2}$ and $\epsilon_2 = 10^{-3}$ and their associated steepest descent algorithm is executed considering $\epsilon_1 = 10^{-4}$ and $\epsilon_2 = 10^{-6}$.

In addition to this stopping criterion, with the aim of limiting the computational time, we have fixed the maximum number of functional evaluation to 50000 for each run (which can be considered, in industrial application cases, as a high number [137, 135]). If at the end of the algorithm (2.4.1) is not satisfied, we consider that the algorithm has failed.

As specified in Section 2.3, we recall that the particular MSA implementations presented previously are adapted to non-negative functions (or with a known lower bound value). So, the benchmark functions h_0 with negative values have been transformed by adding them a real number C_{h_0} high enough to get a non-negative function. Here, in order to obtain a stopping criterion (2.4.1) comparable to the one used in [116, 122], we have considered $C_{h_0} = 2|h_0^*|$.

Due to the stochastic aspect of the considered algorithms, each benchmark problem has been solved 100 times. All experiments have been performed on a Pentium 4 with 3.4 Ghz and 2 Gb of RAM using Matlab script language.

Remark. The considered MSA implementations performs better if the minimum value $h_0^* = 0$ (see Section 2.3.1). So, as h_0^* is known, we could have minimized $h_0 - h_0^*$ instead of h_0 . However, in industrial applications, this information is generally not available [62, 138, 136]. Thus, we have decided to not use this information. Otherwise, we would obtain even better results.

2.4.2 Results

The average number of functional evaluations needed by the optimization algorithms to solve the benchmark problems are presented in Table 2.1. The percentage of runs satisfying the stopping criterion (2.4.1) are reported in Table 2.2.

As we can observe on those tables, the SDMSA-1L algorithm shows the worst success percentages for various benchmark functions. This is due to the fact that the search of a 'good' initial condition is only linear and should fail if the search direction is not adequate.

SDMSA-2L and SDMSA-3L give better results. Both versions present a success percentage of 100 in all cases, which are equivalent to C-GRASP, with a reasonable number of functional evaluations comparing to the DTS (of the same order, lower or higher depending on the case). These good results can be explained by the secant method chosen to perform the linear search: as specified in Section 2.3.1, this method is particularly well adapted to have a quick idea of the functional behavior (for instance, the monotonicity of the function in a direction), then the steepest descent algorithm performs an efficient refined optimization. However, the Shubert and Easom benchmark cases are the two functions which give the highest evaluation numbers. This can be justified by their structure: they present a high number of local minimum with small attraction basins and do not have a convex hull structure which make the global minimum attraction basin difficult to find by using the secant method. However, even in those two cases the success percentage is higher than using DTS. From a general point of view, SDMSA-2L and SDMSA-3L are equivalents, this point out the fact that the two-layers structure is enough to perform a

Function	DTS	C-GRASP	GA	SDMSA-1L
Branin	212	59857	1304	215
Easom	223	189630	40125	2996
Goldstein-Price	230	29	465	463
Shubert	274	82363	7748	416
Hartmann-3	438	20743	1119	564
Hartmann-6	1787	79685	4418	766
Rosenbrock-2	254	1158350	3918	1542
Rosenbrock-5	1684	6205503	43604	4420
Rosenbrock-10	9037	20282529	44557	5612
Shekel-(4,5)	819	5545982	37328	3220
Shekel-(4,7)	812	4052800	36046	2648
Shekel-(4,10)	6828	4701358	40217	3336
Zakharov-5	1003	959	24988	291
Zakharov-10	4032	3607653	40489	593

Function	SDMSA-2L	SDMSA-3L	HBMSA	GMSA
Branin	140	130	489	252
Easom	7085	7427	16504	3488
Goldstein-Price	312	425	1292	439
Shubert	4274	4125	12687	1270
Hartmann-3	347	415	1052	425
Hartmann-6	369	591	5681	1054
Rosenbrock-2	945	1210	2568	1675
Rosenbrock-5	1777	1866	11513	43972
Rosenbrock-10	2657	2545	21151	44828
Shekel-(4,5)	1123	1944	9540	6991
Shekel-(4,7)	936	1483	14780	4619
Shekel-(4,10)	3159	2416	6299	1637
Zakharov-5	152	145	648	2674
Zakharov-10	309	292	2110	20719

Table 2.1: Average number (considering the runs satisfying the stopping criterion (2.4.1)) of functional evaluations needed by the optimization algorithms to solve the benchmark problems.

Function	DTS	C-GRASP	GA	SDMSA-1L
Branin	100	100	100	100
Easom	82	100	100	12
Goldstein-Price	100	100	100	100
Shubert	92	100	100	47
Hartmann-3	100	100	100	100
Hartmann-6	83	100	100	100
Rosenbrock-2	100	100	100	95
Rosenbrock-5	85	100	96	100
Rosenbrock-10	85	99	95	100
Shekel-(4,5)	57	100	97	66
Shekel-(4,7)	65	100	96	37
Shekel-(4,10)	52	100	96	30
Zakharov-5	100	100	100	100
Zakharov-10	100	100	100	100
Function	SDMSA-2L	SDMSA-3L	HBMSA	GMSA
Branin	100	100	100	100
Easom	100	100	48	100
Goldstein-Price	100	100	100	100
Shubert	100	100	91	100
Hartmann-3	100	100	100	100
Hartmann-6	100	100	95	100
Rosenbrock-2	100	100	100	100
Rosenbrock-5	100	100	97	92
Rosenbrock-10	100	100	64	81
Shekel-(4,5)	100	100	66	96
Shekel-(4,7)	100	100	57	98
Shekel-(4,10)	100	100	55	97
Zakharov-5	100	100	100	100
Zakharov-10	100	100	100	100

Table 2.2: Percentage of runs satisfying the stopping criterion (2.4.1).

global search of the initial condition. Therefore, SDMSA-2L and SDMSA-3L present a good alternative to DTS and C-GRASP.

For the HBMSA implementation, the number of functional evaluation is higher and the success percentage is lower than SDMSA-2L and SDMSA-3L. This is normal taking into account that the heavy ball optimization algorithm introduce perturbation that allows to have a better exploratory character of the algorithm (the core algorithm can 'jump' the attraction basins) but in counterpart decreases the algorithm precision (we can pass inside the global minimum attraction basin without stopping). Thus, this version of the algorithm should be preferred in cases of analyzing the behavior of a function without the necessity to have a precise approximation of the optimum.

Finally the GMSA implementation present interesting characteristics. Its success percentage ratio is generally equivalent the GA and better than DTS. Furthermore, the number of functional evaluation is significantly smaller than to the GA alone (with a ratio from 1 to 10 time faster, depending of the case). Therefore, this algorithm is a good alternative to classical GAs, in cases where the gradient of the functional is difficult to compute.

2.5 Conclusions and Perspectives

In this Chapter, a new multi-layers semi-deterministic method based on line search has been introduced. This algorithm provides suitable initial conditions for existing minimization algorithms. A particular implementation, adapted to the minimization of non-negative functions, has been presented and coupled with various core optimization algorithms. Numerical results show that this coupling upgrade the performance of the considered algorithms, presenting a good alternative to other heuristic algorithms, such as DTS and C-GRASP. Those results have been obtained without using the fact that for all studied benchmark problems we know the solution. If we would have used this information, we could have obtained even better results.

Currently, we are implementing our method with other kind of core optimization algorithms (such as Quasi-Newton Methods [174] and GAs different from the one presented here [105]). Additionally, we are developing parallel schemes for those MSAs in order to improve their computational efficiency and range of application.

A Matlab version of the algorithms presented here has been implemented in the free optimization package "Global Optimization Platform", which can be downloaded at

<http://www.mat.ucm.es/momat/software.htm>

As a final remark, we note that two of the algorithms presented in Section 2.4.1, namely SDMSA-2L and GMSA, have been applied to solve several industrial problems involving cost functions with a high computational complexity and having numerous local minima, such as: optical fiber synthesis [135], shape optimization of microfluidic mixers [137], temperature and pollution control in a bunsen flame [62], portfolio risk management [136], control problem of the Burgers equation [138], shape optimization under aerodynamic and acoustic constraints for internal and external flows [201]. Thus, in next Chapters we apply SDMSA-L2 and/or GMSA to solve the optimization problems rising from my new research works.

Chapter 3

Shape Optimization with Geometric Parameterization

Abstract - *In this Chapter, we present various industrial design problems which involve the shape optimization of the considered device. Those problem have been solved by applying a methodology similar to the one presented in my PhD thesis [134]. From a general point of view, the shape is parameterized with scalar geometrical variables (such as angles, lengths, location of interpolation points) and the optimization problem is solved by considering one of the MSA presented in Chapter 2. Here, we consider four particular design problems:*

- *In Section 3.1, we are interested in the design of a microfluidic mixer based on hydrodynamic focusing which is used to initiate the folding process (i.e., changes of the molecular structure) of a protein by diluting a protein solution to decrease its denaturant concentration to a given value in a short time interval we refer to as mixing time. Our objective is to optimize this mixer by choosing suitable shape and flow conditions in order to minimize its mixing time. To this end, we first introduce a numerical model that enables computation of the mixing time of a considered mixer. To reduce the computational time needed to solve our design problem, this model is implemented in both full three-dimensional (3D) and simplified two-dimensional (2D) versions; and we analyze the ability of the 2D model to approximate the mixing time predicted by the 3D model. Then, we define a mixer optimization problem and solve it using the GMSA presented in Section 2.3.3. We study the behavior and verify the robustness of the optimized result by performing an extensive sensitivity analysis on its parameters and geometry.*
- *In Section 3.2, we describe how to tackle new challenging coastal engineering problems related to beach erosion with a shape optimization approach. The method modifies the shape of the sea bottom, by the mean of a geotextile tube, in order to reduce beach erosion effects. Global optimization is shown to be necessary as the related functionals have several local minima. We describe the physical model used, the proposed protection devices against beach erosion and real case applications. The SDMSA-L2, introduced in Section 2.3.1, is used for the search of the optimum properties of the geotextile tube.*

- In Section 3.3, we focus on the modeling, simulation and shape optimization of a dispersive bioreactor in which a substrate is degraded by a microbial ecosystem in an non homogeneous environment. Two different modeling approaches are used in order to obtain a low computational model to quickly evaluate the behavior of our bioreactor. The first one is based on coupled spatial and time dependent PDEs. The second one, obtained by optimization, is based on two interconnected systems of ODEs with coefficient calibrated by using the first model. Preliminary results obtained by considering the GMSA are presented.
- In Section 3.4, we focus on the design of code division multiple access filters (used in data transmission) composed of a particular optical fiber called sampled fiber Bragg grating (SFBG). More precisely, we consider an inverse problem that consists in determining the effective refractive index profile (i.e., shape) of a SFBF that produces a given reflected spectrum. In order to solve this problem, we use various algorithms introduced in Chapter 2. The results obtained with the SDMSA-L2 are compared, in term of complexity and final design, with those given by a genetic algorithm (the method generally considered in the literature for designing SFBGs).

3.1 2D/3D modeling and optimization applied to the design of a fast hydrodynamic focusing microfluidic mixer for protein folding.

3.1.1 Introduction

Proteins are composed of chains of amino acids which can assume complex three-dimensional (3D) structures. Protein folding refers to the processes by which inactive proteins (unfolded chains of amino acids) acquire the 3D shapes (called folded) enabling them to perform a wide range of biological functions [23, 240]. The applications of protein folding in research and industry are numerous, including: drug discovery, DNA sequencing and amplification, molecular diagnostics and food engineering (see, for instance, Refs. [130, 242]). Protein folding can be initiated, for instance, by using photochemical initiation [144], changes in temperature and/or pressure [130, 139] or changes in chemical potential (such as concentration of a chemical specie) [220]. All these techniques provide perturbations of a protein conformational equilibrium [23], necessary to begin folding. The folding techniques based on rapid changes in concentration of chemical species are among the most versatile [181].

The original concept of a micromixer based on diffusion from (or to) a hydrodynamically focused stream was first proposed by Brody et al. in Ref. [36]. As shown in Figure 3.1, this kind of mixer is composed of three inlet channels and a common outlet channel. It is symmetric with respect to its center channel. In the center inlet channel a mixture of unfolded proteins and a chemical denaturant is injected, whereas in the two side inlet channels a background buffer is introduced. The objective is to rapidly decrease the denaturant concentration in order to initiate protein folding in the outlet channel [71]. Since the publication of Brody et al., there have been significant advances in this field.

As summarized by Hertzog et al. [118, 119] and Yao and Bakajin [275], these include reduction in consumption rate of reactants, methods of detection, fabrication and, the most important improvement, reduction of the so called mixing time (i.e., time needed to reach a required denatruant concentration threshold). Indeed, the lower is the mixing time, the higher is the proportion of folded proteins in the outlet stream. For example, while the original mixer of Brody et al. [36] showed mixing times greater than $10\ \mu\text{s}$ (given the mixing measures used here), Hertzog et al. [118] obtained mixing times of $1.2\ \mu\text{s}$. Furthermore, Hertzog et al. [118, 119] and Yao and Bakajin [275] pointed out the importance of 3D flow effects and flow inertia in the designs of these mixers but, due to computational limitations, they considered only 2D flow models.

In this Section, we present both 2D and 3D modeling for the optimization of the shape and flow conditions of a particular hydrodynamic focused microfluidic mixer. Our objective is to improve a specified mixing time of this device taking into account that, currently, the best mixer designs exhibit mixing times of approximately $1.0\ \mu\text{s}$ [118, 275]. To do so, we first introduce a mathematical model which computes mixing time for a given mixer geometry and injection velocities. We develop 2D and 3D versions of this model in order to study the ability of the 2D model to approximate key results of the 3D model. Then, we define the considered optimization problem based on the 2D model. We note that our 2D model is more complex than the one presented in Refs. [118, 137], as it includes new variables such as both the angle of inlet channels near the intersection and inlet flow velocities. This problem is solved by considering the GMSA presented in Section 2.3.3. Finally, using the 3D model, we perform an extensive uniformity and sensitivity analysis of the optimized mixer's mixing time regarding geometric variations including: (i) a study of the impact of variations due to the fabrication process (such as, width and depth variations) on the mixing time; and (ii) the identification of which parameters have the most influence on the mixing time.

This Section is organized as follows: Section 3.1.2 introduces the 2D and 3D models used to compute the mixing times. Section 3.1.3 describes the numerical experiments carried out during this work: a comparison of the models, the optimization process and an extensive sensitivity analysis. Lastly, Section 3.1.4 presents our optimized design results and compare these to published studies [118, 137].

3.1.2 Microfluidic mixer modeling

Here, we detail the mathematical models used to perform both optimization process and sensitivity analysis. More precisely, in Subsection 3.1.2, we define the 2D and 3D models which describe the denaturant concentration distribution of the mixer. In Subsection 3.1.2, we introduce the mixer parameterization determining its shape and flow conditions. Finally, in Subsection 3.1.2, we show how mixing time is computed.

Note that the type of model and numerical approach used here to predict mixing times for given geometry and flow conditions has been validated experimentally in previous studies, including Refs. [118, 119, 275].

Mathematical Model

We consider the microfluidic hydrodynamic focusing mixer introduced in detail in Section 3.1.1.

Let Ω_{3D} be the domain defined by the mixer shape in 3D. A typical representation of Ω_{3D} is depicted in Figure 3.1. The mixer geometry has two symmetry planes that can be used to reduce the simulation domain. Therefore, it is only necessary to study a quarter of the mixer, denoted by $\Omega_{3D,s}$ and represented in dark gray in Figure 3.1. Furthermore, $\Omega_{3D,s}$ can be approximated considering a 2D projection, as suggested in other works [118, 54, 221]. A representation of this projection, denoted by $\Omega_{2D,s}$, is shown in Figure 3.2.

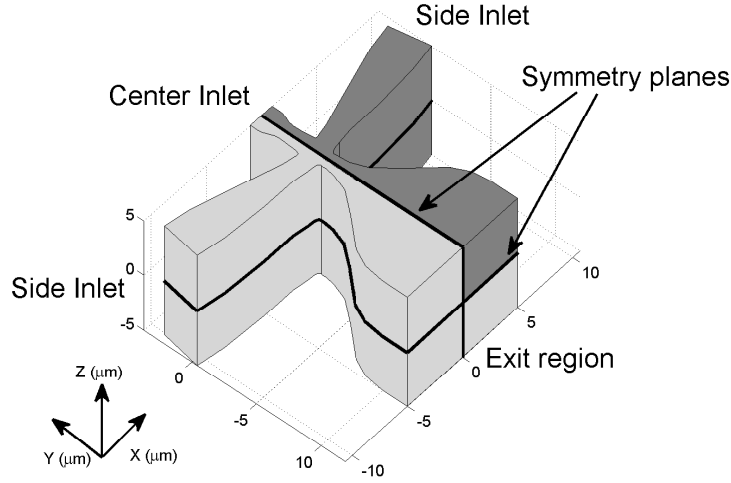


Figure 3.1: Typical domain representation of the microfluidic mixer geometry considering the 3D model: in dark gray we represent the domain $\Omega_{3D,s}$ used for numerical simulations. The geometry's symmetry planes are highlighted and labeled.

For the sake of simplicity, the system of coupled equations introduced below and describing the distribution of the denaturant concentration in the mixer is defined only for the 2D case. The 3D model can be obtained easily by extruding the domain $\Omega_{2D,s}$, the equations and the boundary conditions with the considered mixer depth (i.e, mixer length in the Z-axis) [86].

In order to simplify the notations, we introduce $\Omega = \Omega_{2D,s}$. In the boundary of Ω , denoted by Γ , we define: Γ_c the boundary representing the center inlet; Γ_s the boundary representing the side inlet; Γ_e the boundary representing the outlet; Γ_{w1} the boundary representing the wall defining the lower corner; Γ_{w2} the boundary representing the wall defining the upper corner; Γ_a the boundary representing the Y-axis symmetry. A geometrical representation of these boundaries is given in Figure 3.2.

We assume the mixer liquid flow is incompressible [119]. Thus, the concentration distribution of the denaturant is described by using the incompressible Navier-Stokes equations coupled with the convective diffusion equation [189]. Since we do not need the behavior of the device during its transient set up, only steady configurations are considered. More precisely, we approximate the flow velocity and the denaturant concentration

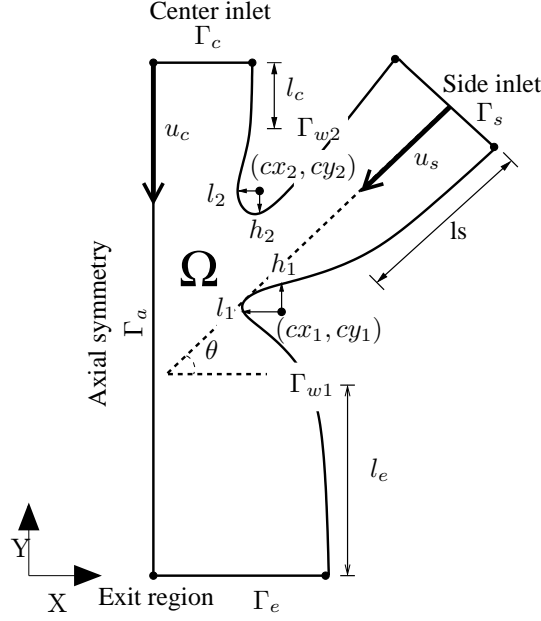


Figure 3.2: Typical representation of the domain $\Omega_{2D,s}$ and parameterization of the microfluidic mixer considered for the optimization process.

distribution by considering the solution of the following system of equations [118, 119]:

$$\begin{cases} -\nabla \cdot (\eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^\top) - p) + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} = 0 & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega, \\ \nabla \cdot (-D \nabla c) + \mathbf{u} \cdot \nabla c = 0 & \text{in } \Omega, \end{cases} \quad (3.1.1)$$

where c is the denaturant normalized concentration distribution, \mathbf{u} is the flow velocity vector (m s^{-1}), p is the pressure field (Pa), D is the diffusion coefficient of the denaturant in the background buffer ($\text{m}^2 \text{s}^{-1}$), η is the denaturant dynamic viscosity ($\text{kg m}^{-1} \text{s}^{-1}$) and ρ is the denaturant density (kg m^{-3}).

System (3.1.1) is completed by the following boundary conditions:

For the flow velocity \mathbf{u} :

$$\begin{cases} \mathbf{u} = 0 & \text{on } \Gamma_{w1} \cup \Gamma_{w2}, \\ \mathbf{u} = -u_s \text{para}_1 \mathbf{n} & \text{on } \Gamma_s, \\ \mathbf{u} = -u_c \text{para}_2 \mathbf{n} & \text{on } \Gamma_c, \\ p = 0 \text{ and } (\eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^\top)) \mathbf{n} = 0 & \text{on } \Gamma_e, \\ \mathbf{n} \cdot \mathbf{u} = 0 \text{ and } \mathbf{t} \cdot (\eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^\top) - p) \mathbf{n} = 0 & \text{on } \Gamma_a, \end{cases} \quad (3.1.2)$$

where u_s and u_c are the maximum side and center channel injection velocities (m s^{-1}), respectively; para_1 and para_2 are the laminar flow profiles (parabolas for the 2D case and paraboloids of revolution for the 3D case) equal to 0 in the inlet border and unity in the inlet center [189]; and (\mathbf{t}, \mathbf{n}) is the local orthonormal reference frame along the boundary.

For the concentration c :

$$\begin{cases} \mathbf{n} \cdot (-D\nabla c + c\mathbf{u}) = -c_0\mathbf{u} & \text{on } \Gamma_c, \\ c = 0 & \text{on } \Gamma_s, \\ \mathbf{n} \cdot (-D\nabla c) = 0 & \text{on } \Gamma_e, \\ \mathbf{n} \cdot (-D\nabla c + c\mathbf{u}) = 0 & \text{on } \Gamma_{w1} \cup \Gamma_{w2} \cup \Gamma_a, \end{cases} \quad (3.1.3)$$

where $c_0 = 1$ is the initial denaturant normalized concentration in the center inlet. We note that the first equality in (3.1.3) corresponds to the inward denaturant flux in the center inlet channel and the third equality to the convective flux leaving the outlet channel.

Mixer parameterization

We first introduce the parameterization used to describe the mixer shape Γ . We consider several constraints related to the mixer microfabrication process [119, 257, 157]: (i) the desired structural strength of the device requires a maximum angle θ at the intersection channels of $\pi/3$; (ii) the depth of the mixer is set to $10 \mu\text{m}$ to avoid clogging issues, to account for the resolution limits of confocal microscopy (used to measure experimentally the mixing time) and to mitigate the effects of the top and bottom walls on mixing dynamics; (iii) the width of the side and center channel nozzles (i.e., the length of Γ_c and twice the length of Γ_s , respectively) are set to $2 \mu\text{m}$ and $3 \mu\text{m}$, respectively; and (iv) the mixer maximum length (i.e, length in the X-axis) and the mixer maximum height (i.e, length in the Y-axis) are set to 24 and $30 \mu\text{m}$, respectively.

Taking these limitations into account, the mixer shape is described by rational Bézier curves and two ellipsoids. The latter are denoted as ellipsoids 1 and 2, where part of the ellipsoid 1 joins, in Γ_{w1} , the outlet and side channels, and part of the ellipsoid 2 joins, in Γ_{w2} , the center and side channels. These curves are determined by the following parameters (see Figure 3.2 for their geometrical representation), suitably bounded to avoid non-admissible shapes (i.e., shape with intersected curves): the angle $\theta \in [0, \pi/3]$ between Γ_c and the direction normal to Γ_s ; the length of the center inlet channel $l_c \in [2.5 \mu\text{m}, 5 \mu\text{m}]$; the length of the side inlet channel $l_s \in [1 \mu\text{m}, 9 \mu\text{m}]$; the length of the outlet channel $l_e \in [0.1 \mu\text{m}, 20 \mu\text{m}]$; the coordinates of the center of the ellipsoid i , with $i = 1, 2$, (cx_i, cy_i) , where $cx_1 \in [0.8 \mu\text{m}, 3 \mu\text{m}]$, $cy_1 \in [l_e \mu\text{m}, l_e + 2 \mu\text{m}]$, $cx_2 \in [0.8 \mu\text{m}, 0.9 \mu\text{m}]$ and $cy_2 \in [cy_1 + 1 \mu\text{m}, cy_1 + 3 \mu\text{m}]$; the radius l_i in the X-axis of the ellipsoid i , with $i = 1, 2$, satisfies $l_i \in [0 \mu\text{m}, (cx_i - 0.5) \mu\text{m}]$; the radius h_i in the Y-axis of the ellipsoid i , with $i = 1, 2$, h_i , satisfies $h_1 \in [0 \mu\text{m}, (cy_2 - cy_1 - 1) \mu\text{m}]$ and $h_2 \in [0 \mu\text{m}, (cy_2 - cy_1 - 1 - h_1) \mu\text{m}]$.

In addition to those parameters, we also consider the maximum injection velocities u_s and u_c as design variables. Furthermore, in order to maintain laminar flow and to avoid secondary flows in the outlet channel, such as Dean vortices [25, 59], we constrained the typical flow Reynolds Re to less than 15 [275, 189, 157]. We define $Re = \rho u_s L / \eta$, where $L = 3 \mu\text{m}$ is the side channel nozzle width. This implies that $u_s \leq \eta Re / \rho L \text{ m s}^{-1}$. Moreover, in practice, u_c should be at least 10 times lower than u_s to ensure a good mixing between fluids [118]. Therefore, we impose that $u_s \in [0, \eta Re / \rho L] \text{ m s}^{-1}$ and $u_c = p \times u_s$, where $p \in [0.001, 0.1]$.

Thus, the set of parameters defining a particular mixer design is denoted by

$$\varphi = \{u_s, p, \theta, l_c, l_s, l_e, cx_1, cy_1, l_1, h_1, cx_2, cy_2, l_2, h_2\} \in \Phi,$$

where $\Phi = \Pi_{i=1}^{14}[\underline{\Phi}(i), \overline{\Phi}(i)] \subset \mathbb{R}^{14}$ is the admissible space; and $\underline{\Phi}(i) \in \mathbb{R}$ and $\overline{\Phi}(i) \in \mathbb{R}$ are the upper and lower constraint values of the i -th parameter in φ described previously, respectively.

Mixing time

In this work, the mixing time is defined as the time required to change the denaturant normalized concentration of a typical Lagrangian stream fluid particle situated in the symmetry streamline at depth $z = 0 \mu\text{m}$ (halfway between the top and the bottom walls) from $\alpha \in [0, 1]$ to $\omega \in [0, 1]$ [118, 119, 275, 137]. We remark that the choice of α and ω has a great impact on the mixing time. This choice is influenced by several factors, such as the type of denaturant [71]. For example, α is set by the minimum denaturant concentration for which we can be confident the protein stays unfolded, while ω is set by the maximum concentration for which we can be confident it folds.

Thus, the mixing time of a particular mixer described by the parameters $\varphi \in \Phi$, and denoted by J_{2D} for the 2D case and J_{3D} for the 3D case, is computed by:

$$J_{iD}(\varphi) = \int_{c_\omega^\varphi}^{c_\alpha^\varphi} \frac{dy}{\mathbf{u}^\varphi(y)}, \quad (3.1.4)$$

where i is the dimension of the problem (i.e., $i=2$ or 3); \mathbf{u}^φ and c^φ denote the solution of System (3.1.1)-(3.1.3), in its i D version, when considering the mixer defined by φ ; and c_α^φ and c_ω^φ denote, for the 2D case (3D case, respectively), the points situated along the symmetry streamline (the streamline defined by the intersection of the two symmetry planes $z = 0 \mu\text{m}$ and $x = 0 \mu\text{m}$, respectively) where the denaturant normalized concentration is α and ω .

Remark 1 *As suggested in [221], the mixing time should be measured not only in the central streamline but also in other streamlines originated from the center channel boundary Γ_c . The idea is to study the non-uniformity of the mixing time across the focused stream, since in practice, the mixing time is measured by a confocal microscope probe volume taking into account various fluid streamlines [157]. For instance, as done in [275], we could minimize the mean value of the mixing times of the considered streamlines instead of function (3.1.4). However, this approach is computationally expensive (in our case, evaluating the streamlines takes 3 min versus 35 s for solving the 2D model). Thus, taking into account the large number of numerical experiments carried out during this work (see Section 3.1.3) and that this approach has been previously considered in literature (see [118]), we decide to minimize (3.1.4) and study the mixing time uniformity of the optimized mixer as a posteriori analysis.*

3.1.3 Numerical Experiments

In this Section, we first introduce our numerical implementation of the 2D and 3D models. Then, we describe the numerical experiments accomplished to compare both

models, to optimize the mixer and to analyze the validity and robustness of the optimized result.

Numerical implementation of the model

The numerical versions of both 2D and 3D models, presented in Section 3.1.2, are implemented by coupling Matlab scripts (www.mathworks.com) with COMSOL Multiphysics 3.5a models (www.comsol.com). More precisely, to compute a numerical solution of System (3.1.1)-(3.1.3), we consider a Finite Element Method (FEM) with Lagrange P2-P1 elements to stabilize the pressure and to satisfy the Ladyzhenskaya, Babouska and Brezzi stability condition. The 2nd-order Lagrange elements model the velocity and concentration components, while linear elements represent the pressure. The Navier-Stokes equations are solved using Galerking Least Square streamline and crosswind diffusion methods in order to prevent numerical oscillations. The convective diffusion equation is solved by considering an upwind scheme. We use a Direct Damped Newton method to solve the corresponding linear systems. Finally the mixing time, defined by Equation (3.1.4), is estimated by considering the solutions of previous FEM model and a trapezoidal approximation of the integral. A complete description of those techniques can be found in Ref. [100].

The computational experiments are carried out in a 2.8Ghz Intel i7-930 64bits computer with 12GB of RAM. For the 2D simulations described in Sections 3.1.3 and 3.1.3, we use a Delaunay mesh with around 6000 elements. In that case, a single evaluation for J_{2D} requires about 35 s. The 3D simulations, conducted during Sections 3.1.3 and 3.1.3, are performed with a Delaunay mesh containing 13000 elements. Each evaluation of J_{3D} takes approximatively 30 min.

Comparison between 2D and 3D models

First, a comprehensive computational study is carried out to determine if both 2D and 3D models yield similar mixing times when they are evaluated with the same set of parameters. Indeed, if both models have a similar behavior, the computational effort for solving the optimization problem presented in Section 3.1.3 can be reduced by using the 2D model instead of the 3D one (see Section 3.1.3).

Let $\{\varphi_i\}_{i=1}^{100}$ be a set of 100 mixers randomly generated in Φ by considering a uniform distribution. For each one of them, we evaluate: the concentration distribution $c_{2D}(\varphi_i)(x, y)$, the velocity field $\mathbf{u}_{2D}(\varphi_i)(x, y)$ and the mixing time $J_{2D}(\varphi_i)$ for the 2D model; the concentration distribution $c_{3D}(\varphi_i)(x, y, 0)$ and the velocity field $\mathbf{u}_{3D}(\varphi_i)(x, y, 0)$ in the plane $z = 0 \mu\text{m}$, and the mixing time $J_{3D}(\varphi_i)$ for the 3D model. Then, we compute the relative difference, in percentage, between the solutions obtained by the 2D and 3D models as following:

$$100 \frac{|J_{2D}(\varphi_i) - J_{3D}(\varphi_i)|}{J_{2D}(\varphi_i)}, \quad (3.1.5)$$

$$\frac{100}{\int_{\Omega} dx dy} \int_{\Omega} \frac{|c_{2D}(\varphi_i)(x, y) - c_{3D}(\varphi_i)(x, y, 0)|}{|c_{2D}(\varphi_i)(x, y)|} dx dy, \quad (3.1.6)$$

$$\frac{100}{\int_{\Omega} dx dy} \int_{\Omega} \frac{\|\mathbf{u}_{2D}(\varphi_i)(x, y) - \mathbf{u}_{3D}(\varphi_i)(x, y, 0)\|_2}{\|\mathbf{u}_{2D}(\varphi_i)(x, y)\|_2} dx dy. \quad (3.1.7)$$

Additionally, for each of those quantities, we calculate the mean, minimum and maximum values regarding the 100 generated mixers.

Finally, we want to know if the 2D model preserve the same order of mixing time between two particular mixers as the 3D model. To do so, we sort the previous 100 mixers by their J_{2D} value, and analyze in which proportion the order is maintained regarding J_{3D} .

Design problem

The objective is to design a microfluidic mixer described by parameters $\varphi \in \Phi$, where $\Phi \subset \mathbb{R}^N$ and $N = 14$, that minimizes the mixing time function J_{2D} defined in Section 3.1.2. Thus, the associated optimization problem can be written as:

$$\min_{\varphi \in \Phi} J_{2D}(\varphi). \quad (3.1.8)$$

In order to solve Problem (3.1.8), we use GMSA introduced in Chapter 2 (with B_1 , $t_{l_1} = 50$, $N_g = 10$, $N_p = 10$, $p_c = 0.45$ and $p_m = 0.35$). We denote by φ_o the result obtained at the end of the optimization process.

Analysis of the optimized result

In order to study the behavior of φ_o , we perform the following experiments:

- Firstly, we want to check the improvements obtained by our optimized mixer. Additionally, we want to study the behavior of φ_o when considering the 3D model. Indeed, some important effects cannot be appreciated with the 2D model, as for example, the impact of upper and lower mixer walls on the velocity field or possible effects of certain secondary flows. To this aim, we analyze the mixing time, the shape, the final concentration and the velocity field of φ_o by considering both 2D and 3D models and compare them to other results found in literature [118, 275, 137].
- Secondly, as was specified in Remark 1, we are interested in studying the non-uniformity of the mixing times across the focused stream associated to our optimized mixer φ_o . To this aim, we consider the 3D model and 100 streamlines, denoted by $(sl_{i,j})_{i,j=1}^{10}$, starting from a finite set of points, which are denoted by Σ_{Γ_c} , in Γ_c . Here, $\Sigma_{\Gamma_c} = \{x_{(i,j)} | i = 1, \dots, 10 \text{ and } j = 1, \dots, 10\}$ where $x_{(i,j)} = (\frac{i}{10}0.9\mu\text{m}, \frac{j}{8}0.75\mu\text{m})$. In the previous definition, the maximum coordinate in the X-axis (i.e., $0.9 \mu\text{m}$) has been selected in order to avoid particles too close from the wall Γ_{w1} , and the maximum coordinate in the Z-axis (i.e., $0.75 \mu\text{m}$) have been chosen as the depth of field for the considered confocal microscope is assumed to be $1.5 \mu\text{m}$ [118]. Those streamlines are numerically approximated by considering an explicit Euler scheme and the velocity vector \mathbf{u} obtained by solving System (3.1.1)-(3.1.3) [130].

For each streamline $sl_{i,j}$, we compute the associated mixing times, denoted by $t_{sl_{i,j}}$, on a similar way than Equation (3.1.4). More precisely, $t_{sl_{i,j}}$ is defined as the time required by a Lagrangian protein stream fluid particle to travel from $c_{90}^{sl_{i,j}}$ to $c_{30}^{sl_{i,j}}$, where $c_{90}^{sl_{i,j}}$ and $c_{30}^{sl_{i,j}}$ denote the points situated in $sl_{i,j}$ with a concentration of 90% and 30%, respectively. Next, we compute the mean mixing time value, denoted by $\overline{t_{sl}}$, defined as

$$\overline{t_{sl}} = \frac{\sum_{i,j=1}^{10} \omega_{i,j} t_{sl_{i,j}}}{\sum_{i,j=1}^{10} \omega_{i,j}}, \quad (3.1.9)$$

where $\omega_{i,j}$ denotes the velocity of a particle in the streamline $sl_{i,j}$ at its initial position in Γ_c . This choice of weight coefficient is motivated by the fact that the probe volume used to measure experimentally the mixing time receive more particles from the streamlines with highest velocities. The maximum and standard deviation values of those weighted mixing times is also studied.

Due to the fact than the depth of the mixer is 10 times larger than the center channel width, the mixing time variations in the Z-axis direction should be negligible in comparison to the variations in the X-axis [118]. Thus, we perform a more extensive uniformity analysis in the X-axis direction, by considering 100 streamlines, denoted by $(sl_{i,z=0})_{i=1}^{100}$, in the plane $z = 0$ starting from Γ_c . The methodology is the same than the one introduced previously. These results will be compared with the ones presented in [118].

- Thirdly, we want to perform a simple sensitivity analysis on φ_o . This study consists on randomly perturb all the parameters of φ_o by taking uniform variations in a range of $[-\alpha\%, +\alpha\%]$ of their value. This perturbation process is repeated 100 times. For each perturbed mixer, denoted by $\varphi_{p,\alpha}$ with $p = 1, \dots, 100$, we compute $J_{3D}(\varphi_{p,\alpha})$ and compare it to $J_{3D}(\varphi_o)$ through the relation:

$$100 \frac{|J_{3D}(\varphi_o) - J_{3D}(\varphi_{p,\alpha})|}{J_{3D}(\varphi_o)}. \quad (3.1.10)$$

Then, we compute the mean, minimum and maximum values of Equation (3.1.10) regarding the 100 perturbed mixers. The objective of the sensitivity analysis is twofold. On the one hand, we want to know if φ_o is close to a local minimum of the design problem when considering the 3D model. To this aim, we apply small perturbations of amplitude $\alpha = 1\%$ and focus on the mixers with lower mixing time than φ_o . On the other hand, we want to analyze the robustness of φ_o (i.e., the variations on its mixing time) when the parameters are strongly perturbed. For this case, perturbations of amplitude $\alpha = 5\%, 10\%$ and 20% are taking into account [118].

- Finally, we want to determine which parameter is the most influential one in the mixing time. To do so, each parameter of φ_o is modified by $\pm 5\%, \pm 10\%$ and $\pm 20\%$ of its predicted optimized value. Then, we compare the mixing time obtained with the modified mixers, denoted by φ_{mod} , to the optimized design by using a percent variation relation similar to the Equation (3.1.10).

Furthermore, we have also studied several of the “fixed” parameters in our design constraints. In particular, we have analyzed the effects of changes in the central, side and exit channel width. To this aim, each channel width have been subsequently varied $\pm 0.2\mu\text{m}$, $\pm 0.5\mu\text{m}$ and $\pm 1\mu\text{m}$, and their corresponding mixing times have been compared to $J_{3D}(\varphi_o)$. The effects of the mixer depth have been also analyzed, since during the fabrication process, it is expected depth variations of approximately $\pm 1\mu\text{m}$ [119]. Then, we have computed the mixing time for mixers generated by considering the set of parameter φ_o and a depth of $9\mu\text{m}$ and $11\mu\text{m}$. Those mixing times are compared to $J_{3D}(\varphi_o)$ through a prelation similar to Equation (3.1.10). For the sake of completeness, mixers with depth equal to $8\mu\text{m}$ and $12\mu\text{m}$ have been also studied.

3.1.4 Numerical Results

Here, we present the results obtained by performing the experiments described in Section 3.1.3 when considering the denaturant introduced in Section 3.1.4. In particular, Section 3.1.4 studies the comparison between the 2D and 3D models, and Section 3.1.4 analyzes the behavior of the optimized mixer.

Considered denaturant

During this work, we have considered guanidine hydrochloride (GdCl) as the denaturant [71]. Indeed, GdCl is a Chaotropic agent which is frequently used for protein folding.

The thermophysical variables of GdCl can be approximated as those of water. More precisely, its density is $\rho = 1010\text{ kg m}^{-3}$ and its dynamic viscosity is $\eta = 9.8 \times 10^{-4}\text{ kg m}^{-1}\text{ s}^{-1}$. Furthermore, the diffusion coefficient of GdCl in the background buffer (assumed to be similar to water) is $D = 2 \times 10^{-9}\text{ m}^2\text{ s}^{-1}$. According to those coefficients and the restriction $Re = \rho v L / \eta \leq 15$ introduced in Section 3.1.2, the maximum side injection velocity is $u_s \leq 7\text{ m s}^{-1}$.

Finally, the values of α and ω in Equation (3.1.4) adapted to GdCl are considered here as 0.9 and 0.3, respectively [118, 137].

Comparison between the 2D and 3D models

In Table 3.1, we report the mean, minimum and maximum relative percent variation values between the solutions of Equations (3.1.5)-(3.1.7) obtained by the 2D and 3D models. The mean percent variation in the mixing time is 15.3%, showing that the 2D model approximates, in a reasonable way, the mixing time predicted by the 3D model. As can be seen, the largest percent variations are obtained in the velocity field, with a mean percent variation of 18.7% versus only 9.8% for the concentration distributions. From these results, we may conclude that the ability of the 2D model to match the solutions (e.g., mixing time or concentrations) of the 3D model is sufficient.

An important feature of the 2D model is its ability to preserve the same order of mixing time between two different mixer designs as the 3D model (i.e., if $J_{2D}(\varphi_1) \leq J_{2D}(\varphi_2)$ then $J_{3D}(\varphi_1) \leq J_{3D}(\varphi_2)$, for most of φ_1 and $\varphi_2 \in \Phi$). For this purpose, we represent in Figure 3.3 the 2D mixing time of the 100 mixers generated previously, sorted according

	Mean	Minimum	Maximum
Mixing time	15.3	1.2	56.7
Concentration	9.8	0.2	18.7
Velocity field	18.7	3.4	32.9

Table 3.1: Mean, minimum and maximum percent variation (%) of the mixing time, concentration distribution in the plane $z = 0 \mu\text{m}$ and velocity field in the plane $z = 0 \mu\text{m}$ obtained when considering the 100 microfluidic mixers randomly generated during the 2D-3D comparison experiments detailed in Section 3.1.3.

to their 2D mixing time, as well as their respective 3D mixing times. The 3D mixing time order is preserved in 72% of the cases. In addition, when the order between two consecutive mixers is not conserved, the difference in their mixing times is, on average, about 12% which can be considered as a low value.

All these results suggest that the optimization process can be performed by using the 2D model instead of the 3D one.

Analysis of the optimized mixer

As said previously, the optimization problem (3.1.8) has been solved by considering the 2D model and the GMSA. The convergence history of the GMSA is depicted in Figure 3.4. The number of evaluations of J_{2D} used by GMSA was about 6000 and the optimization process spent around 60 h. Notice that, as a single evaluation of J_{3D} takes approximatively 30 min, solving the same optimization problem with the 3D model could require more than 125 days, which is not a reasonable time.

The values of φ_o are reported on Table 3.2. The shape of the optimized microfluidic mixer, its concentration distribution and the concentration evolution of a particle in its central streamline, obtained with the 2D model, are depicted in Figure 3.5. The maximum side injection velocity is $u_s = 5.2 \text{ m s}^{-1}$, whereas the maximum central injection velocity is $u_c = 0.038 \text{ m s}^{-1}$. The Reynolds number Re , defined in Section 3.1.2, is around 9. The mixing time associated to this mixer is about $0.10 \mu\text{s}$. This value is 10 times lower than the mixing times achieved by previous mixer designs with the same 2D model [118, 119, 137]. In those works, the mixing times were each greater than 1μ . We attribute this improvement mainly to three factors: (i) the angle θ of the inlet side channels, whose value is about $\pi/5$ radians (this angle was fixed to 0 in Refs. [118, 119, 137]); (ii) the width of the mixing region (i.e., the area, defined by $(x, y) \in [0, 2] \times [14, 19] \mu\text{m}$ and depicted in Figure 3.6, where both fluids are mainly mixed) which reaches a minimum value of about $1.1 \mu\text{m}$; and (iii) the choice of adequate injection velocities. Indeed, as can be observed in Figure 3.6, the shape of both corners Γ_{w1} (stretched along the Y-direction) and Γ_{w2} (sharply pointed wedge pointing roughly along the Y-axis) yields a reduced channel width of $1.1 \mu\text{m}$ near $y = 16.5 \mu\text{m}$, where the maximum velocity rises up to 26 m s^{-1} , which helps to accelerate the mixing time. Moreover, we note that the optimized value of θ is consistent with the experimental work described in Ref. [275]. Such a study highlights the importance of inclined side channels to prevent strong centripetal accelerations in the

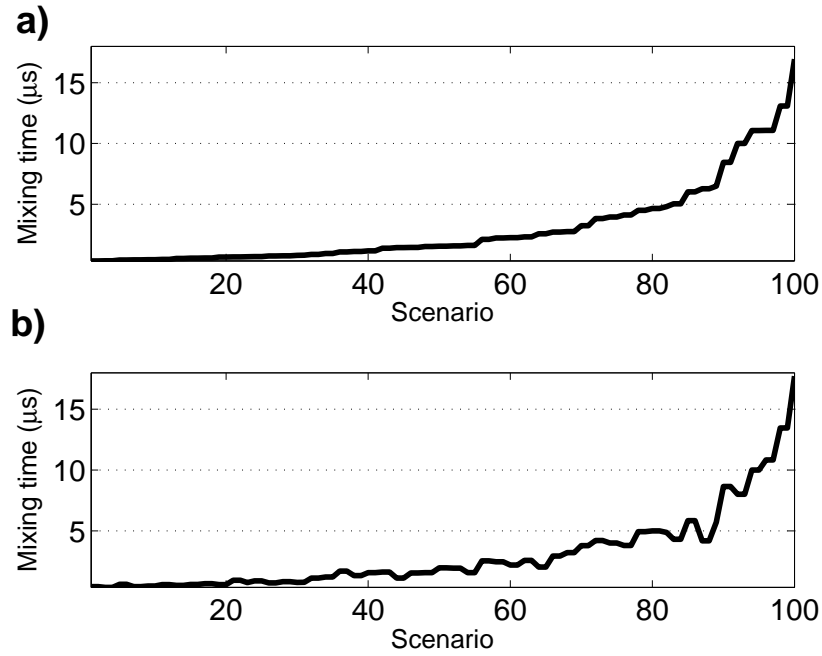


Figure 3.3: Mixing times of the 100 microfluidic mixers (called Scenarios), randomly generated during the 2D versus 3D comparison process and sorted considering the 2D mixing time, as computed when considering **a)** the 2D model and **b)** the 3D model.

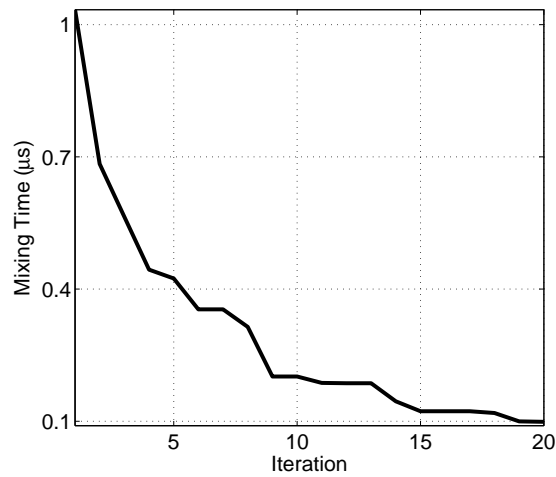


Figure 3.4: Convergence history of the GMSA obtained during the optimization process.

Parameter	u_s	p	θ	l_c	l_s	l_e	cx_1
Value	5.2	7.3×10^{-3}	0.6	2.5	9.1	16.3	1.1
Parameter	cy_1	l_1	h_1	cx_2	cy_2	l_2	h_2
Value	16.6	0.5	0.3	0.9	18.9	0.1	1.1

Table 3.2: Values of the optimized microfluidic mixer parameters presented in Section 3.1.4.

inlet side channel streams, which deteriorate the mixer performances.

We also compute the mixing time for this optimum design using the 3D model. To this aim, we extrude the 2D optimal shape (see Figure 3.7) and we evaluate it using the 3D model. The predicted mixing time is also around $0.10 \mu s$ (the difference in mixing times with the 2D model is lower than about 4%). In Figure 3.6, we show the concentration and velocity amplitude distributions achieved by the 2D and 3D models in the $z = 0 \mu m$ midplane and considering the mixing region. Both solutions exhibit similar characteristics, although we can observe some differences, especially in the velocity field; for the 3D case: (i) Re is around 7 (instead of 9 in the 2D case); and (ii) the maximum velocity reached near $y = 16.5 \mu m$ is $19 m s^{-1}$ (instead of $26 m s^{-1}$ in the 2D case). Figure 3.7-a) shows selected streamlines generated by the 3D velocity field. As we can observe on this figure, the fluid remains laminar. The concentration distribution as well as the velocity amplitude distribution, both obtained with the 3D model, are shown in Figure 3.7-b) and -c), respectively. These two figures exhibit the so-called wall effect [133], since the no-slip condition at the mixer walls results in low velocity values near those walls. This can also be observed in Figure 3.6, where the X-Z slices of the concentration and velocity amplitude distributions are depicted. These low velocities result in higher denaturant concentrations in these regions. However, the mixer is designed to be relatively insensitive to such wall effects by maintaining a relatively large depth of $10 \mu m$ (while having a minimum channel width of $1.1 \mu m$ in the mixing region).

The study of the mixing time non-uniformity presented in subsection 3.1.3 and applied to our optimized mixer yield that the mean mixing time value, defined by (3.1.9), is $0.34 \mu s$ with a standard deviation of $0.17 \mu s$. As expected, the maximum mixing time value is reached at the streamline further from the central streamline, i.e. $sl_{10,10}$, with a value of $1.43 \mu s$. The mixing time and weighted mixing time distributions, according to initial the position the considered streamline in Γ_c , is presented in Figure 3.8-a). As can be observed on this figure, the mixing time is more sensitive to changes in the X-axis, with a mean standard deviation of $0.18 \mu s$ in this direction, than in the Z-axis, with a mean standard deviation of $0.01 \mu s$. Furthermore, we can see that the weighted mixing time of streamlines close from the mixer wall Γ_{w1} (i.e., at position $> 0.8 \mu m$) is low whereas the regular mixing time is high (i.e., $> 1 \mu s$). This is due to the fact that, those particles have a small velocity and, thus, are not taken into account by the probe volume with a high frequency. For the 100 streamlines ($sl_{i,z=0}$) $_{i=1}^{100}$ case, the mean mixing time is $0.32 \mu s$ and the standard deviation is $0.16 \mu s$. The maximum mixing time is $1.41 \mu s$. The spatial distribution for the X-axis is depicted in Figure 3.8-(b). The mixing time and weighted mixing time distributions, regarding the initial position of the streamlines, is presented in Figure 3.8-b). Those results can be, in part, compared to those observed during the

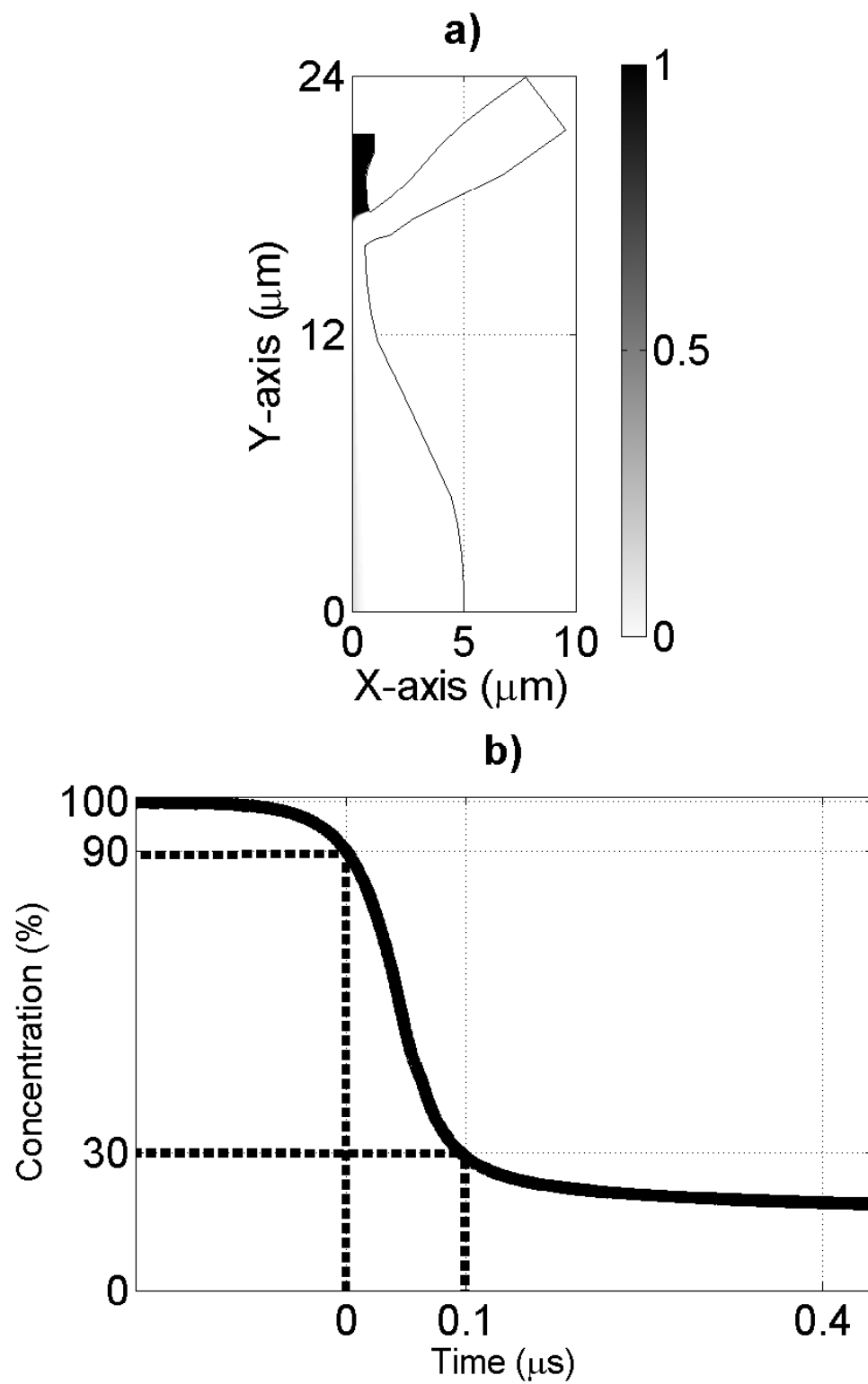


Figure 3.5: Optimized mixer simulated with the 2D model: **a)** shape of the optimized mixer with a superposed color plot of the denaturant concentration distribution and **b)** the time evolution of the denaturant concentration of a particle in the symmetry streamline.

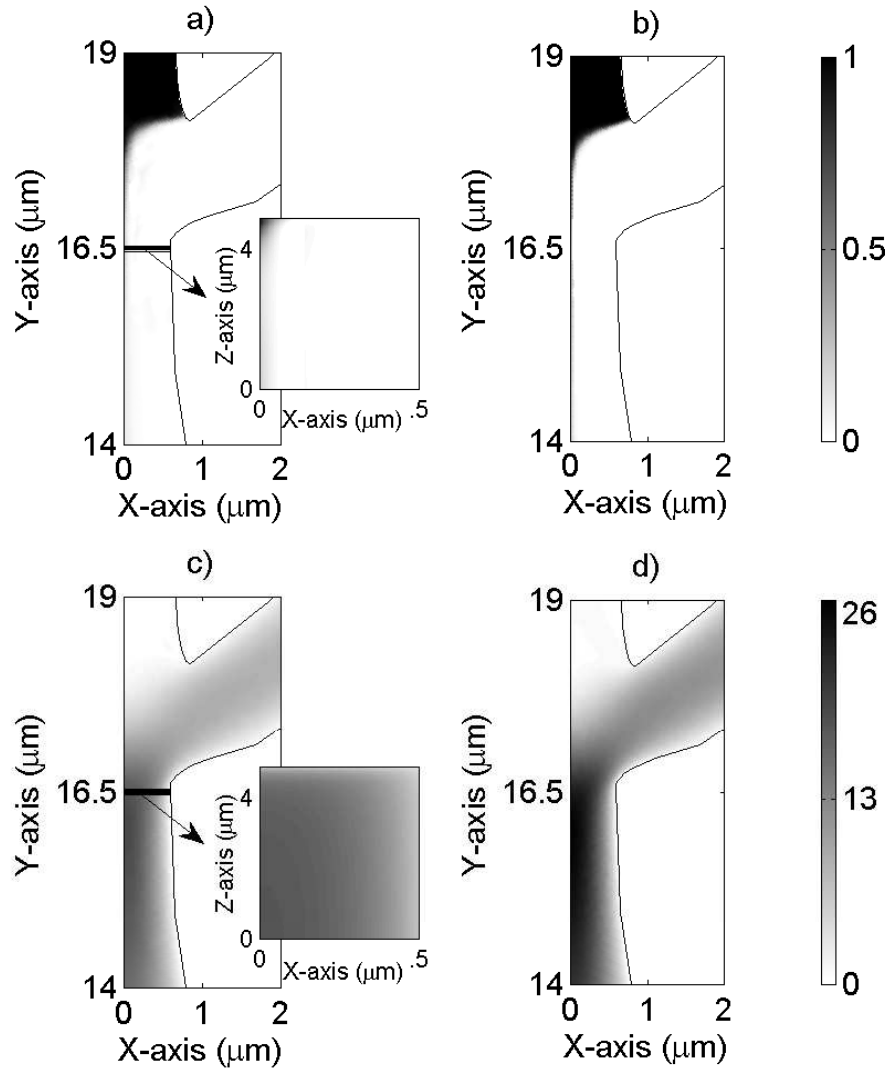


Figure 3.6: Comparison of the solutions obtained in the mixing region with the optimized mixer considering the 3D model (subfigures **a)** and **c)**) and the 2D model (subfigures **b)** and **d)**). **a)** and **b)** show denaturant concentration distributions while **c)** and **d)** plot velocity amplitude distributions in the symmetric plane $z = 0 \mu\text{m}$. For the 3D case, the figure also shows, in the inset detail views, the X-Z plane slices of the concentration and velocity amplitude distributions at the plane defined by $y=16.5\mu\text{m}$ (represented with a horizontal black lines).

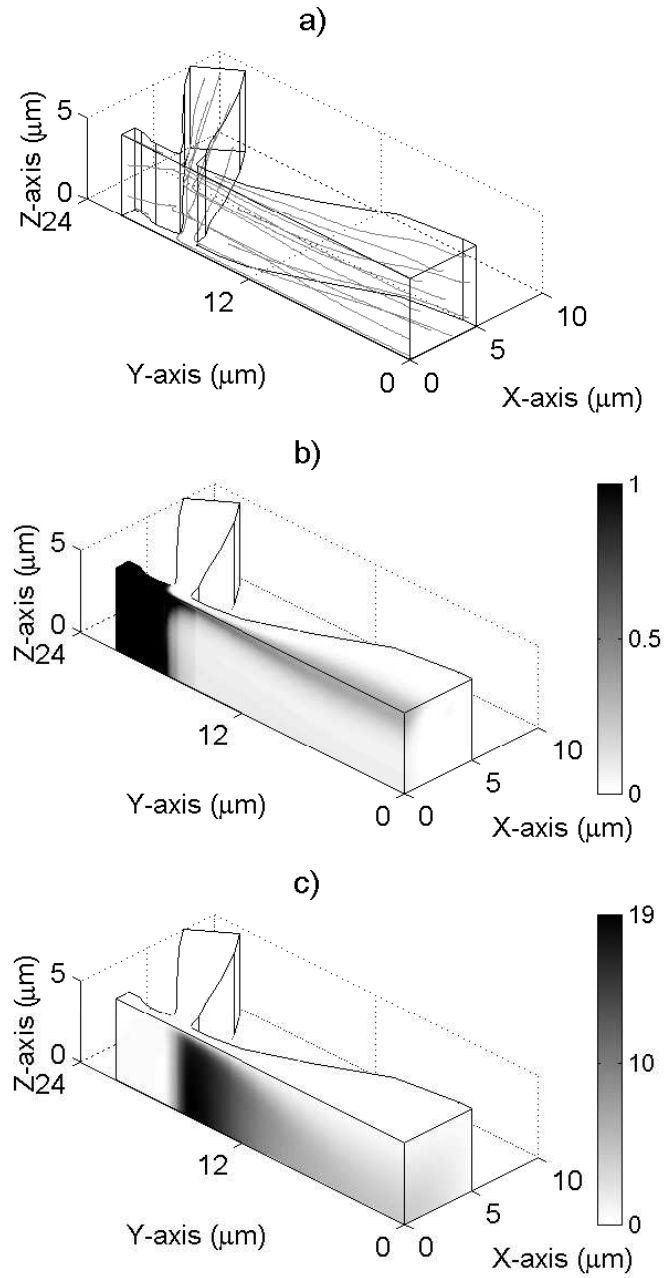


Figure 3.7: Sample results from the optimized mixer design as simulated with the 3D model. Shown are isometric views of the shape of the optimized mixer with representation of **a)** selected streamlines generated by considering the velocity field, **b)** the concentration distribution, and **c)** the velocity amplitude distribution.

Maximum Amplitude	Mean	Minimum	Maximum
1%	1.2	0.1	2.1
5%	6.1	0.6	12.7
10%	13.7	2.5	29.2
20%	21.7	4.4	45.7

Table 3.3: Mean, minimum and maximum percent variation (%) in the mixing time obtained by considering the 3D model and by perturbing randomly all the parameters of φ_o with a maximum amplitude of 1%, 5%, 10% and 20% of their initial value.

study presented in [118]. The mixer presented in that work exhibited a mean mixing time, considering streamlines in the plane $z = 0$ and a formula similar to (3.1.9), of $3.1 \mu\text{s}$ with a standard deviation of $1.5 \mu\text{s}$. The maximum mixing time value was $10 \mu\text{s}$, obtained for the streamline closest from Γ_{w1} . Thus, the optimized mixer presented here behaves much better, regarding the mixing time uniformity, than the one described in [118].

The results of the sensitivity analysis of the optimized parameters presented in Section 3.1.3 when $\alpha = 1\%, 5\%, 10\%$ and 20% are reported on Table 3.3. We first focus on the case where the amplitude of the perturbations applied to φ_o are lower than 1% . We observe that the optimized mixer has a better mixing time in 77% of the cases. We attribute the imperfect prediction of the 3D optimum to, on the one hand, the lack of precision of the considered GMSA method (due to the high computational time required to evaluate Equation (3.1.4), the number of iteration of the algorithm, and thus its precision, has been restricted) and, on the other hand, the differences between the 2D and 3D models. However, it is important to mention that the mean mixing time variation between the optimized mixer and the mixers with the lowest mixing times, is smaller than 0.4% . This result suggests that the optimized mixer may be considered a solution close to a local minimum of Problem (3.1.8) when the 3D model is used.

Furthermore, as can be seen on Table 3.3-Column *Mean*, the mean percent variation in the mixing time (caused by variations in the input parameters) is proportional to the maximum percent variation of the parameter perturbation. In particular, the mixing times of the perturbed mixers are of the same order than the mixing time of the optimized mixer, suggesting the optimized solution is stable. Additionally, even in the worst case (i.e., Table 3.3-Line 20%-Column *Maximum*), the perturbed mixer still exhibits a mixing time of $0.15 \mu\text{s}$, which is a significant improvement compared to previous mixers proposed in literature [118, 119]. All those results indicate that φ_o is a robust solution for our design problem.

We analyzed the sensitivity of the optimized design by perturbing each of the geometrical and flow parameters. The results are summarized in Table 3.4, where we list the % variation in mixing times resulting from $\pm 5\%$, $\pm 10\%$, and $\pm 20\%$ variations for each of the parameters. The most influential parameters are the injection velocities u_c and u_s , as well as the parameters l_1 and h_2 which define the details of the corner shapes. These results are consistent with trends observed experimentally: namely, that injection velocities and their ratio have a strong influence on mixing time [119, 275]. Moreover, the corner shapes effect a reduction of flow rate which strongly increase fluid velocity and extensional strain in the mixing region. In contrast to these parameters, other parameters defining channels

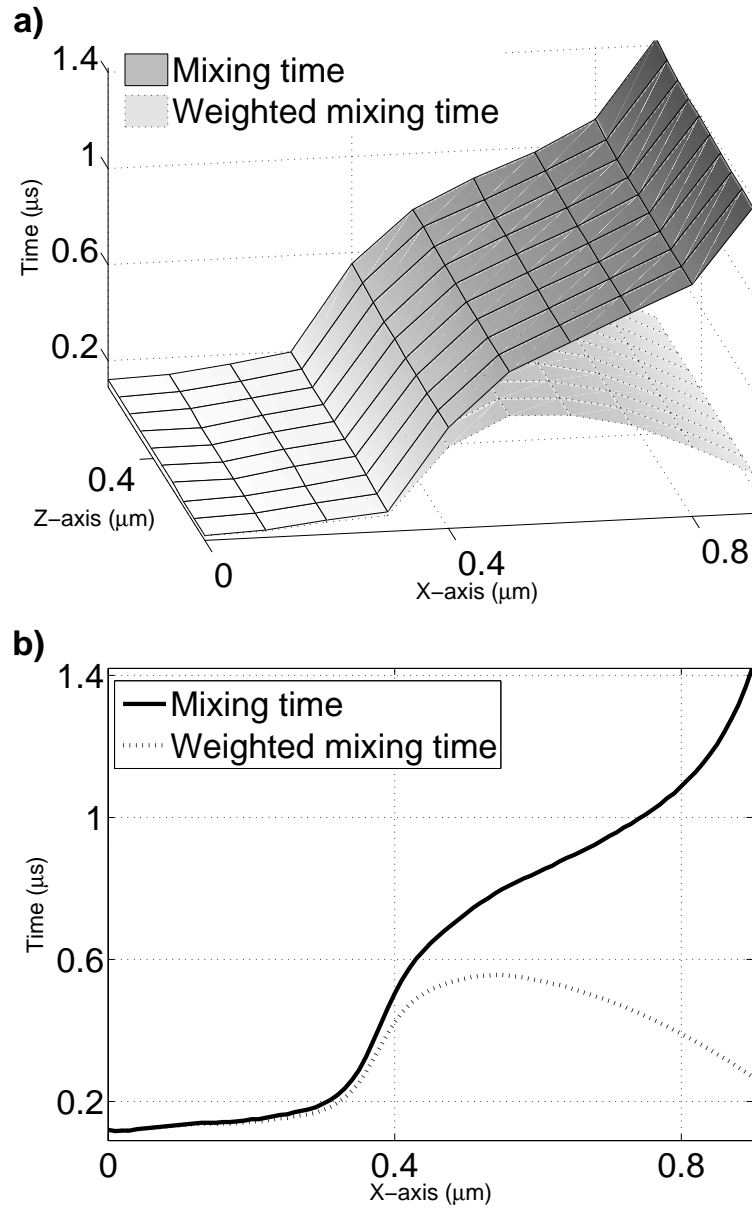


Figure 3.8: Mixing and weighted mixing times (μs) obtained according to the position: **a)** (x, z) and **b)** $(x, 0)$ of the initial particle in Γ_c for the considered streamlines.

Table 3.4: Maximum percent variation (%) in the mixing time obtained by perturbing by $-\alpha\%$ and $+\alpha\%$, with $\alpha = 5, 10$ and 20 , each optimized mixer parameters.

Parameter	l_1	l_2	h_1	h_2	p	u_s	θ
$\pm 5\%$	5.7	1.0	0.5	5.5	5.8	8.3	0.4
$\pm 10\%$	11.7	1.3	1.0	11.3	11.9	17.5	0.8
$\pm 20\%$	26.7	2.8	1.1	23.9	27.4	40.7	0.9
Parameter	cx_1	cx_2	cy_1	cy_2	l_e	l_s	l_c
$\pm 5\%$	0.6	0.3	7.2	1.1	0.2	0.1	1.7
$\pm 10\%$	1.6	0.4	8.0	2.4	0.6	0.2	3.0
$\pm 20\%$	2.4	0.5	14.1	4.6	0.5	0.3	6.5

Table 3.5: Mixing time (in μs) obtained by varying the **central**, **side** and **exit** channel width by $\pm 0.2\mu m$, $\pm 0.5\mu m$ and $\pm 1\mu m$.

Width (μm)	-1	-0.5	-0.2	$+0.2$	$+0.5$	$+1$
central	0.38	0.16	0.10	0.12	0.41	1.02
side	0.16	0.12	0.12	0.10	0.10	0.10
exit	0.11	0.11	0.10	0.10	0.10	0.10

length l_e , l_s and l_c ; the corner centroid cx_1 , cx_2 , cy_1 and cy_2 ; and the angle θ have only a weak influence on the mixing time. This is interesting as it offers flexibility to experimentalists in designing the downstream outlet channel shapes (which help set the relation between Lagrangian time coordinate and spatial location of the proteins as they unfold).

Results regarding the variation of the channel width are summarized on Table 3.5. The data show the width of the central channel is the most sensitive parameter. For example, imperfections in fabrication which increase the central channel width by about $1\mu m$ can result in a 10-fold increase of mixing time to $1\mu s$. This sensitivity is not surprising as this stream width (together with the flow rate ratio) helps control the hydrodynamic focusing of the center stream. For example, effects which tend to decrease center channel stream initially help by decreasing the characteristic time for spanwise (X-direction diffusion to occur). However, eventually decreasing center stream width results in decreased velocities in the center inlet stream nozzle. The latter moves low concentration regions upstream into the low velocity regions of nozzle; this increases the residence time proteins spend at intermediate concentrations. The latter effect is discussed by Hertzog et al. [119].

The results obtained by varying the optimized micro-mixer depth are summarized in Table 3.6. As shown, perturbations of $\pm 1\mu m$, generate reasonable percent variations in the mixing time between 6% and 13%. As described previously, this indicates that errors in the mixer depth due to fabrication process not strongly affect mixing performance for these relatively deep ($\sim 10\mu m$) mixers. Note the highest channel depth yields the lowest mixing time ($0.09\mu s$ for a depth equal to $12\mu m$ versus $0.14\mu s$ for the $8\mu m$ depth). This result is expected, since the “wall effect” mentioned in subsection 3.1.4, reduces the mixing

Table 3.6: Mixing time (in μs) obtained with the optimized mixer when considering a depth of 8, 9, 10, 11 and 12 μm , respectively. The percent variation (%) regarding the mixing time obtained with the initial depth of 10 μm is also reported.

Depth (μm)	8	9	10	11	12
Mixing time (μs)	0.14	0.12	0.10	0.10	0.09
Percent variation	34	13	-	6	13

performance near the mixer walls. Again, we see that the optimal mixer design (minimum mixing time) is influenced strongly by challenges in fabrication (namely in achieving high aspect ratio features with deep reactive ion etching). Mixer designs with relatively high channel-depth-to-feature width ratios yield optimal results. For our study, the minimum channel width (near $y = -1.5 \mu\text{m}$) was $1.1 \mu\text{m}$. The nominal depth explored was $10 \mu\text{m}$ and the maximum depth explored was $12 \mu\text{m}$.

3.1.5 Conclusions and Perspectives

We have been explored the design of a particular fast hydrodynamic focusing microfluidic mixer for protein folding. The objective was to reduce the mixing time of this kind of mixer by optimizing the shape (in particular the angle of the side channels) and the injection velocities.

To this aim, firstly, we have introduced a mathematical model used to compute the mixing time of a mixer according to the defined design variables. This model has been presented in 2D and 3D versions. Results show that the 2D and 3D models exhibit similar behaviors regarding the mixing time. Thus, we concluded that the 2D model can be used in the optimization process to greatly reduce the computational time.

Secondly, we defined the optimization problem associated to the design of our device, and solved this using a GMSA. The optimized mixer shows a mixing time of $0.1 \mu\text{s}$, which represented a decrease of a factor 10 compared to previous known mixers.

Finally, we have verified the robustness of the optimized mixer performances by performing an extensive sensitivity analysis on its parameters and geometric variable. It has also allowed us to identify the most influential mixer parameters and to provided recommendations for the fabrication of this mixer.

The next step, as done in previous works [118, 137], should be the validation of our optimized result by fabricating a prototype and performing real experiments.

3.2 Shape optimization of geotextile tubes for sandy beach protection

3.2.1 Introduction

Beach erosion problems bring increasing engineering demand. Indeed, about 70% of world beaches are crossing an erosion phase, 20% are stable and 10% show signs of fattening [222, 223]. Obviously, this has major economical and environmental impacts. Groins, breakwaters and other coastal structures are used to decrease water wave energy or to control sediment flows. The shape of these devices are usually determined using simple hydrodynamical assumptions, structural strength laws or empirical considerations. However, as we will see, these are not fully satisfactory because of secondary effects. Our aim is to take advantage of shape optimization techniques, mainly used in aeronautics [200, 141], to propose new solutions to tackle this problem. This approach is fully innovative in coastal engineering.

Also, efficient global optimization algorithms are necessary to avoid the design to converge to local minima. Indeed, we will see that the related functionals have several local minima. Moreover, the search space is often non-connected. We use the SDMSA-L2 presented in Section 2 to allow global optimization of systems governed by PDEs with a low calculation complexity [134, 137].

This Section is structured as follows. In section 3.2.2, we recall the state of the art on geotextile tubes and we describe their parameterization. Section 3.2.3 presents the flow model used for the water waves propagation. Section 3.2.4 is dedicated to the description of the minimization problem. Finally, Section 3.2.5 displays and discusses optimization results for two beach protection studies in Northwestern Mediterranean sea with the aim of reducing the energy available for sediment transport.

3.2.2 Geotextile tubes and erosion process

Water waves propagating toward the coast are characterized by their height H , their period T and their direction θ . The period does not change during the propagation but the direction and the height may vary when approaching emerged or submerged structures. This is mainly due to scattering phenomenon.

Knowing the height H of the wave is crucial for the study of an erosion problem. Indeed, the suspension of sediments, produced by water wave action, is the main mechanism of erosion process. The suspension of sediments is essentially linked to the associated water wave mechanical energy $E = \frac{1}{8}\rho g H^2$ where ρ is the water density and g the gravity acceleration [60].

Oceanographer observations of erosion show that the water waves can be roughly sorted in two categories according to their height H , below or above a critical value H_{lim} . Basically, those above H_{lim} , mainly present during storms, are erosive. They generate large mechanical energy. On the other hand, when $H < H_{lim}$, waves foster the reconstruction of eroded beaches. In that follows, the first class of water waves will be called erosive and the second class constructive.



Figure 3.9: Secondary effect of emerged groins: erosion has been amplified on one side of the groins (downstream from the longshore drift).

In order to decrease water waves impact along the coastline many structures have been proposed [230]. Until recently the most used are emerged break-waters or groins built with rocks or concrete. However, these techniques are expensive and only short-term solution for the beach protection because they mainly transfer in space the erosion process (figure 3.9 shows the negative impact of emerged groins on erosion. Accretion occurs only upstream from the longshore drift, whereas erosion is amplified downstream. Hence, it is soon necessary to build another groin further downstream... and so on).

Currently, interest goes to a new generation of soft structures having less impact on coastal hydro-sedimentary system [232]. These devices are geotextile tubes, also called geotubes (figure 3.10-**Up**). These geotubes are long cylinders made of synthetic textile and filled up with sand.

This paper discusses shape and location *optimizations* of geotubes for two sites both located on the Northwestern Mediterranean French coast.

The first analysis concerns the protection of a 2.4 km beach located between *Sète* and *Agde* [94] (figure 3.10-**Down**-(Left)). This is an large-scale industrial project under strict feasibility constraints. The *Bas-Rhône Languedoc* Company (BRL) is in charge of the device layout and installation. This company has a great experience of land-use management and development in Languedoc-Roussillon (a french area) in order to perform engineering studies and advisory services. Concerning the studied site, the coastal zone is characterized by a very low tidal excursion and moderate waves. This beach is subjected to severe erosion and the coastline has recorded a shoreward displacement of about 50 m since 1967 with a rate of shore retreat of about 1 m/year since 10 years. In this spot, the critical wave height H_{lim} is about 2 m.

The second situation concerns the protection of the *Le Grau du Roi Le Boucanet* beach

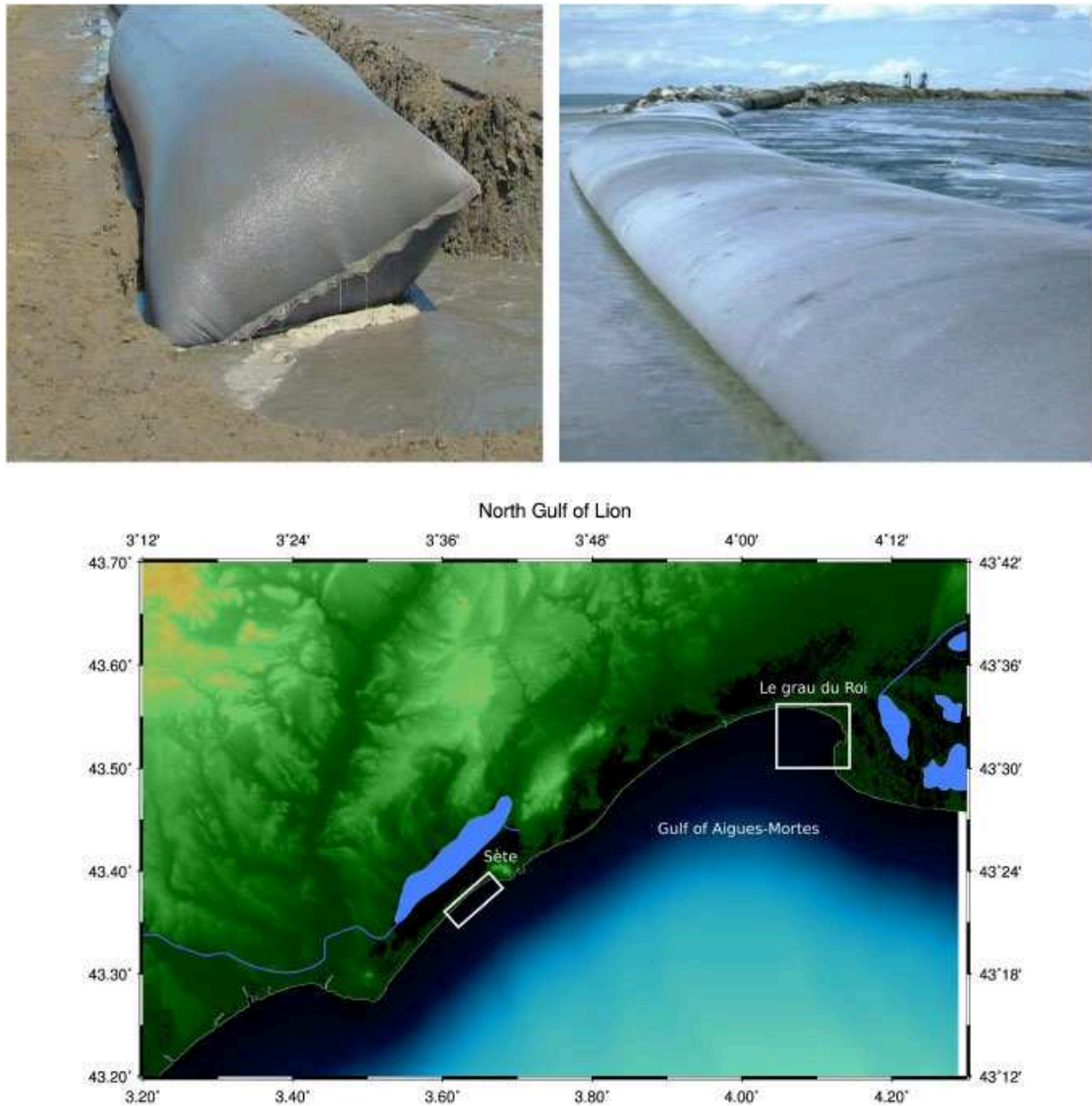


Figure 3.10: **Up-** A geotextile tube before submersion; **Down-** Left: the barred beach between *Sète* and *Agde*, Right: the beach of *Le Grau du Roi Le Boucanet*.

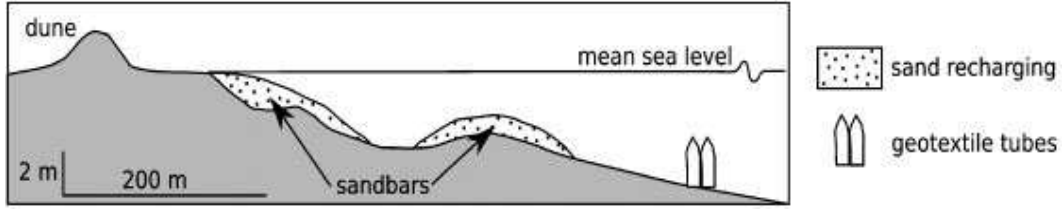


Figure 3.11: The improvement of the considered site

(figure 3.10-Down-(Right)). This problem leads to a higher dimension design problem, as we shall see below.

Geotube parameterization

Along the barred beach between *Sète* and *Agde*, the coastal topobathymetric profile includes two natural sandbars, created under the water wave action. An engineering preliminary study suggested first to restructure the beach and the two natural sandbars by sand recharging and secondly to place two geotubes side-by-side behind the second natural sandbar in order to protect the new beach. In the sequel, we model the two geotubes as a single one twice larger. Figure 3.11 shows a cross-section of the initial topobathymetry, the suggested sand recharging and the location of the two geotubes.

The initial topobathymetric data available for this study consists of series of echo-sounding data obtained from numerous monitorings since 2000 [195]. So, we recreate exactly the real morphodynamic of the spot.

Geotubes will merely be represented by local modification of the topobathymetry. More precisely, in each node of the discrete domain we have a given value for the initial topobathymetry. To add a geotube arbitrarily in the domain, we parameterize its position using a series of N control points in the domain. Cubic splines are used to connect these points. Once the location known, the shape (cross-section) of the geotube is given by a Gaussian function of the form $f(d) = He^{-sd^2}$ where d is the normal distance to the spline. This adds two additional parameters (s, H) for the control of the height and width of the geotubes. Thus, we have in each node a modification of the topobathymetry (see figure 3.12) which accounts for the presence of the geotubes.

3.2.3 Flow model

The propagation of water waves over linear irregular bottom bathymetry and around islands involves many processes such that shoaling, refraction, energy dissipation and diffraction. In this paper, we describe a parabolic weakly nonlinear combined refraction and diffraction model which incorporates all of the effects mentioned above [152, 154]. This model is based on a Stokes expansion of the water wave problem and combines the essential features of the two specific approaches, a mild slope model [28] and a diffraction model [193].

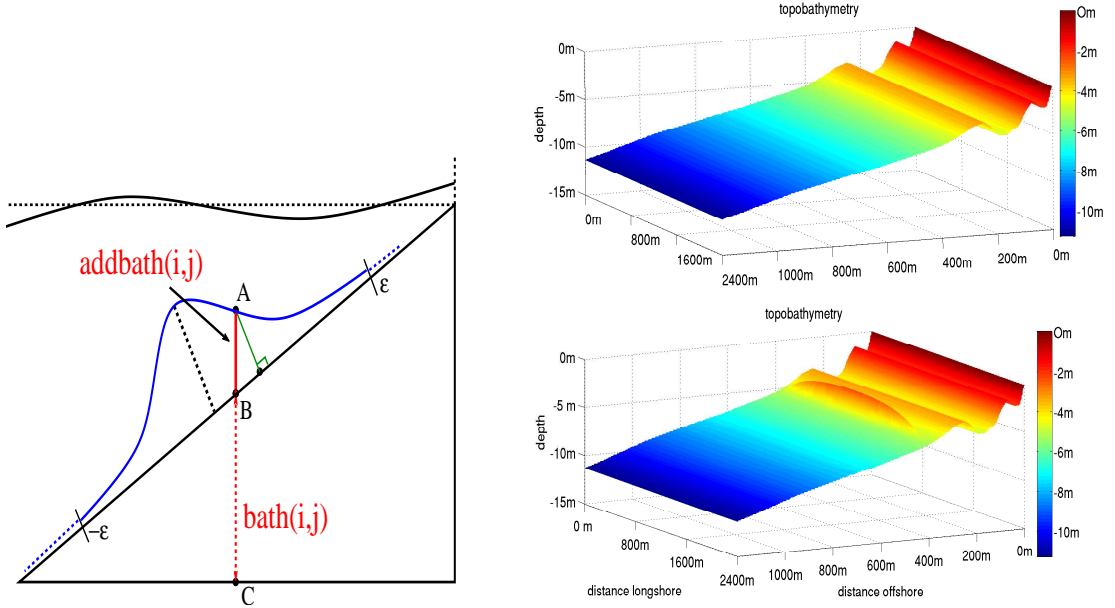


Figure 3.12: **Left:** The profile parameterization for the geotube in an academic linear topography; **Right: Up-** The initial topobathymetry of the barred beach, **Down-** Implementation of a geotube in the topobathymetry. Note that both ends are smoothed by a suitable function.

Nonlinear combined refraction/diffraction model

Kirby & Dalrymple, at the Center for Applied Coastal Research (University of Delaware, US), developed a more general formulation governing waves in a domain with slow but arbitrary depth variations [152]. The following parabolic approximation for refraction/diffraction is obtained,

$$2ikCC_g \frac{\partial A}{\partial x} + 2k(k - k_0)(CC_g)A + iA \frac{\partial(kCC_g)}{\partial x} + \frac{\partial \left(CC_g \frac{\partial A}{\partial y} \right)}{\partial y} - k(CC_g)K'|A|^2A = 0 \quad (3.2.1)$$

where

$$C = \sqrt{\frac{g}{k} \tanh kh} \quad (\text{phase speed}), \quad (3.2.2)$$

$$C_g = C \frac{(1 + \frac{2kh}{\sinh 2kh})}{2} \quad (\text{group velocity}), \quad (3.2.3)$$

$$K' = k^3 \left(\frac{C}{C_g} \right) \frac{\cosh 4kh + 8 - 2 \tanh^2 kh}{8 \sinh^4 kh}, \quad (3.2.4)$$

$h(x, y)$ is the local water depth and g the acceleration of the gravity. The local wave number, $k(x, y)$, is related to the angular frequency of the waves, ω , and the water depth h by the nonlinear dispersion relationship

$$\omega^2 = gk \tanh kh. \quad (3.2.5)$$

k_0 is a reference wave number related to the incoming condition $x = 0$ of the incident wave, $x = 0$ being the offshore boundary.

In the model (3.2.1), $A = A(x, y)$ is the complex amplitude related to the water surface displacement by

$$\eta(x, y) = \text{Re}(A(x, y)e^{ikx}) \quad (3.2.6)$$

and equation (3.2.1) is numerically solved by an implicit finite difference method [153, 154]. On lateral boundaries, open boundary conditions are considered.

This model reproduces water wave propagations over a variable bottom bathymetry under mild slop assumption ($\nabla h/kh \ll 1$). It takes into account shoaling, refraction, energy dissipation and diffraction. However, this model does not account for the reflection phenomenon. This appears, for instance, in water wave propagation in a harbor with vertical emergent structures. Our applications only concern propagation toward sandy beaches. The model has been validated on various experiments [45, 229].

3.2.4 Cost function

In this study, our aim is to optimize the shape of a given geotube and its distance to the coast in order to minimize the energy available for sediment transport in the near-shore zone. We have seen in section 3.2.2 that this cost function can be expressed as a function of the water wave mechanical energy $E = \frac{1}{8}\rho g H^2$ where ρ is the water density and $H = 2A$ [60]. This energy is crucial in the erosion process as it fosters the motion of sediments.

Another important issue for the modified geometry (i.e. after addition of the geotube) is that the geotube should not increase the sea bottom fluid particle velocity¹ shoreward. For a water wave propagation in the x-direction, the stationary bottom orbital velocity U_{orb} is given by [193]

$$U_{orb} = \frac{Agk}{\omega} \cos \mathbf{k} \cdot \mathbf{x} \quad (3.2.7)$$

However, section 3.2.2 and Table 3.7 suggest two main categories of water waves: the constructive and the erosive ones which, as we said, correspond to waves with heights below and above the critical water height $H_{lim} = 2\text{m}$. Hence, for a given direction of propagation θ , the following cost function is considered

$$J_\theta = \frac{\int_D E_{H>H_{lim}} dS}{\int_D E_{H<H_{lim}} dS} + (\|U_{orb}\| - \|U_{orb}^{initial}\|)_+ + \left(\int_D E_{H<H_{lim}} dS - \int_D E_{H<H_{lim}}^{initial} dS \right)_+ \quad (3.2.8)$$

where $(x)_+ = \max(x, 0)$ is a regularized max operator in order to have J_θ differentiable and *initial* is related to the initial unmodified sea bottom. D is the area where the minimization is desirable. Hence, minimizing J aims to maximize the constructive waves and minimize destructive ones (as a low-pass filter for waves) under the constraint that both the orbital velocity energy and the energy produced by constructive waves should always be lower than their initial values (i.e. for the unperturbed region).

¹Also called bottom orbital velocity

	South	South South East	East South East	East
constructive	$H_s = 0.76\text{m}$	$H_s = 0.85\text{m}$	$H_s = 0.85\text{m}$	$H_s = 0.66\text{m}$
water	$T_s = 4.96\text{s}$	$T_s = 5.21\text{s}$	$T_s = 5.21\text{s}$	$T_s = 4.99\text{s}$
waves	$p = 24.66\%$	$p = 22.75\%$	$p = 22.75\%$	$p = 17.5\%$
erosive	$H_s = 2.91\text{m}$	$H_s = 3.233\text{m}$	$H_s = 3.233\text{m}$	$H_s = 3.55\text{m}$
water	$T_s = 7.54\text{s}$	$T_s = 7.78\text{s}$	$T_s = 7.78\text{s}$	$T_s = 8.03\text{s}$
waves	$p = 2.84\%$	$p = 3.25\%$	$p = 3.25\%$	$p = 2.5\%$

Table 3.7: Typical hydrodynamic data used in our simulations . Significant height H_s , mean period T_s and mean frequency of observation p for four significant directions of propagation.

As we are interested by multi-point optimization, we consider the following weighted combination

$$J = \sum p_\theta J_\theta \quad (3.2.9)$$

where p_θ is the observation frequency for a given wave direction. This last point again brings the importance of global minimization tools.

This is also clear from figure 3.13 which shows a sampling of the functional J along one dimension for a situation where the only parameter is the distance of the geotube to the beach. The area D where functional J is computed lies between 100 and 250 meters far from the coastline. The height of the geotube has been set to 3 m, its width to 12 m and the offshore distance sampled between 100 and 750 m seaward for a simple straight geotube. The minimum is obtained for a geotube located at around 350 m from the coastline (we come back to this simulation in section 3.2.5).

3.2.5 Application to beach erosion control

In this section we apply the SDMSA-L2 (with A_2 , $t_0 = 10$, $t_{l_1} = 5$, $t_{l_2} = 5$ and $\epsilon = 0$), presented in Section 2.3.1, to PDE based shape optimization problems arising in the design of beach protection device. We consider two different sites where the beach is either straight or curved (see figure 3.10).

Straight geotube

We first design a straight geotube. This problem arises in the protection of a beach located between *Sète* and *Agde* (Northwestern Mediterranean sea, Languedoc-Roussillon, France) where industrial constraints reduce the number of design parameters to two: the distance from the coast and the height of the tube. The computational domain for the flow is 2400 m longshore and 1200 m shoreward. The mesh size is 1m cross-shore and 5m longshore. The tube is of the same length as the studied costal zone. In that case, as the tube was already pre-designed, the width is fixed to 12 m. The propagation is performed for water waves data available at 1.2 km of the coastline, stemming from French National Center Archive for In Situ Wave Data (e.g. from table 3.7). Recreational sailing and tourism navigation constraints due to local topobathymetry imply that the search space

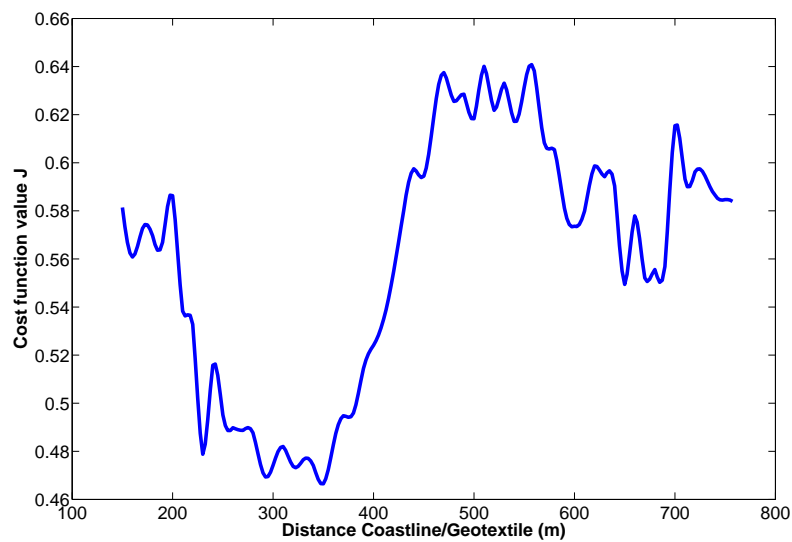


Figure 3.13: Cost function evolution w.r.t to the geotube position. This is difficult to solve with a local minimization algorithm.

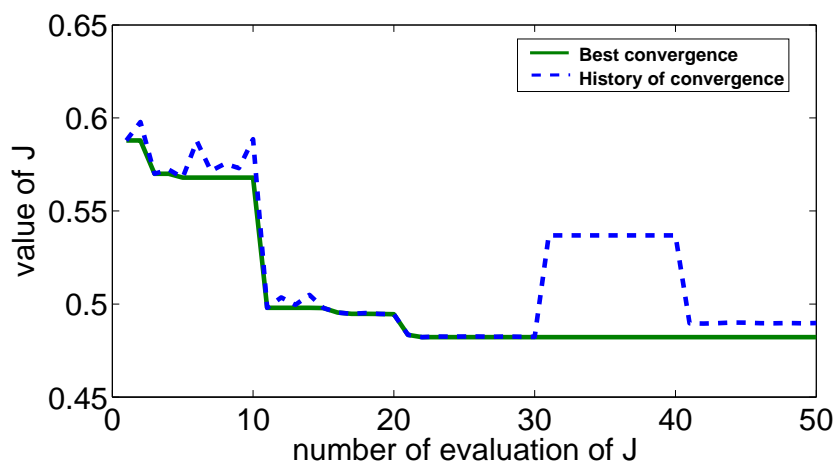


Figure 3.14: Functional values history for the best element found by the core minimization algorithm and the current value of the functional. One sees that the optimization algorithm is not a descent method.

	South	South South East	East South East	East
constructive water waves	==	==	==	==
weights	24.66%	22.75%	22.75%	17.5%
erosive water waves	15% gain	30% gain	16% gain	15% gain
weights	2.84%	3.25%	3.25%	2.5%
overall gain	23%			

Table 3.8: Multi-point optimization: variations of constructive and destructive energies for various incidences θ . (==) means that the quantity is almost unchanged, less than 0.1% variation.

$[100 \text{ m}, 200 \text{ m}] \cup [300 \text{ m}, 850 \text{ m}]$ is disconnected. This because no tube should be placed on the second natural sandy bar (see figure 3.11) to keep a minimum depth of 2 m. Coastal engineering suggests to build devices to control the energy generated in the area between 100 and 250 meters far from the coastline (denoted by D in the cost function definition, see section 3.2.4). This zone corresponds to the gap between the first and the second natural sand bar.

Coastal engineering first guess would suggest to set the geotextile tube around 550 m from the coast to recreate the third natural sand bar which is missing in this site. Available geotubes are 3 m high which leaves an acceptable depth of water to float ships. The optimization procedure locates the geotube at 353 m far from the coast with a height of 2.5 m. This result is confirmed by the sampling of figure 3.13.

Table 3.8 shows that the optimized geotube is inactive for the constructive water waves and reduces the erosive ones. In addition, because of the multi-point problem considered, this is true for all studied propagation directions : the geotube plays its low-pass filter role.

Global functional histories for the current and best element found are shown in figure 3.14. We plot the output of the core minimization algorithm for different initializations provided by our multi-layer construction. Several local attraction basins have been visited. The core minimization algorithm has been called about 50 times. Overall the optimization has required about 700 functional evaluations. Each state evaluation requires about 15 min on a (3 GHz - 1 GB) personal computer. Figure 3.15 shows the water wave height for the protected and unprotected beaches for erosive SSE condition (See Table 3.7).

This shows that a geotube set immediately before the second natural sand bar makes it possible to break the water waves, therefore attenuating their energy in the nearshore zone. More precisely, this study reveals that the geotube must break the water waves sufficiently far away from the coastline but not too far, otherwise the wave recreates itself. In addition, figure 3.16 ensures that this optimized configuration does not increase the bottom orbital velocity compared to the initial configuration, which guarantees that

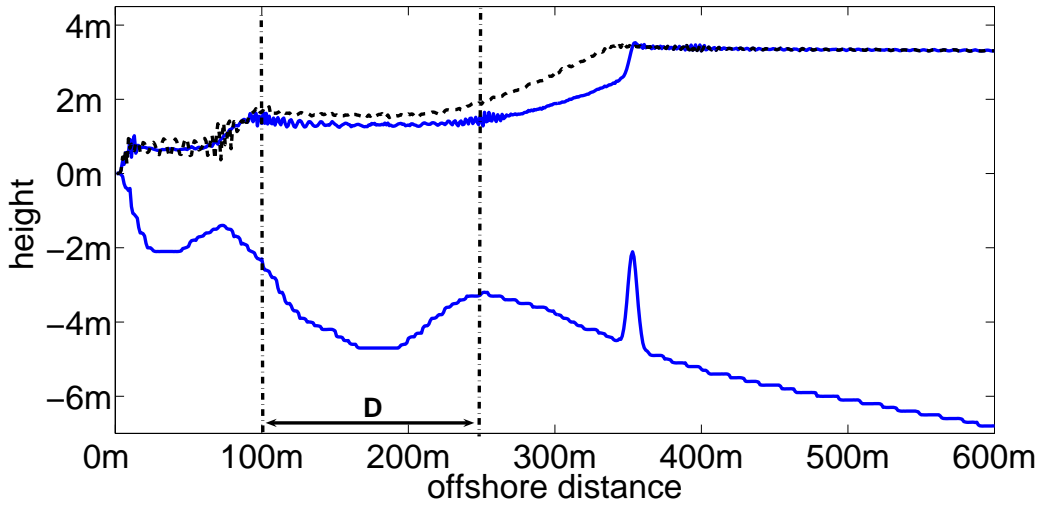


Figure 3.15: Bottom curve: cross-shore topobathymetry. Solid top curve: water height compared to the original height (dashed line).

bottom shear-stress is not amplified (see section 3.2.4).

Curved geotube

The topobathymetry is the one of *Le Grau du Roi Le Boucanet* beach (Northwestern Mediterranean sea, Languedoc-Roussillon, France). The studied area corresponds to a 3 km longshore and 4.4 km seaward zone with a mesh size of 5 m. We consider the design of one geotube defined by 8 control parameters. A spline is defined passing by 3 nodes (this makes 6 parameters). The two remaining parameters are height and width of the tubes, which are let free as well. The length of the tube is required to remain between 500m and 2000m.

The cost function (3.2.8) is minimized in the region D , which lies between 100 et 250 m offshore, depicted in figure 3.17-(Left). For computation time reasons, we only consider a mono-directional south incident water wave with a period $T = 8$ s and an amplitude $A = 3$ m.

The optimized geotextile tube is 1.5 km long with a height of 13 m and a width of 70 m. Its location is represented in figure 3.17-(Right). This optimized configuration reduces drastically the energy available for sediment transport. In this case, the cost function decreases by more than 65% compared to the unmodified topobathymetry. In practice, it is obvious that putting this optimized geotube into construction won't be easy, owing to its size. But in this section, the objective is to highlight the efficiency of the global optimization algorithm.

We display in figure 3.18 the water height H in the whole domain for the two configurations. We observe that, in the region D (close to the east coast), the water height is reduced from 2 m to 1 m by the use of the optimized geotextile tube. Also, we observe an important decrease of the bottom orbital velocity in the optimized configuration (see fig-

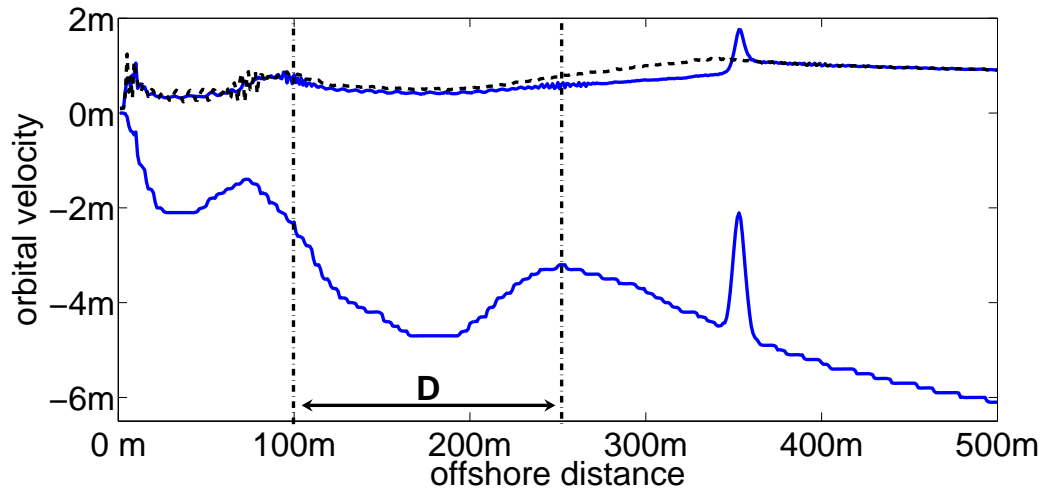


Figure 3.16: Bottom curve: cross-shore topobathymetry. Solid top curve: orbital velocity compared to the velocity on the unprotected beach (dashed line).

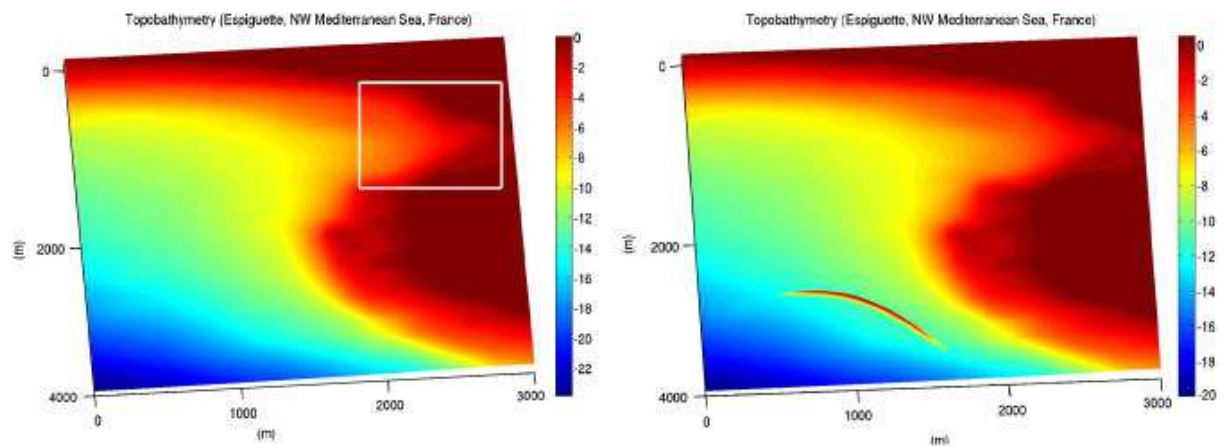


Figure 3.17: (Left): The initial topobathymetry where the white box represents the region D for the cost function computation; (Right): The modified topobathymetry with the optimized geotube.

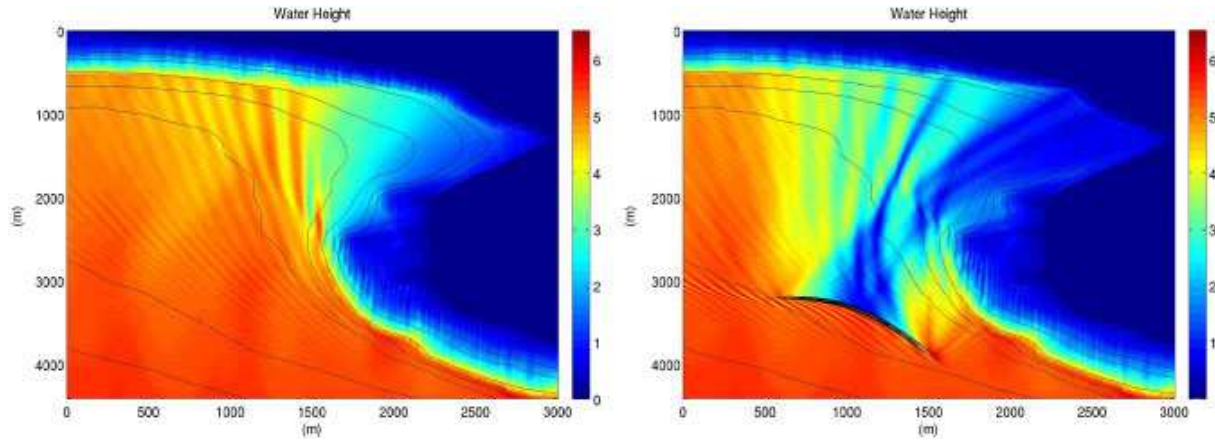


Figure 3.18: The water height H in the whole domain: (Left) for the initial configuration, (Right) for the optimized configuration.

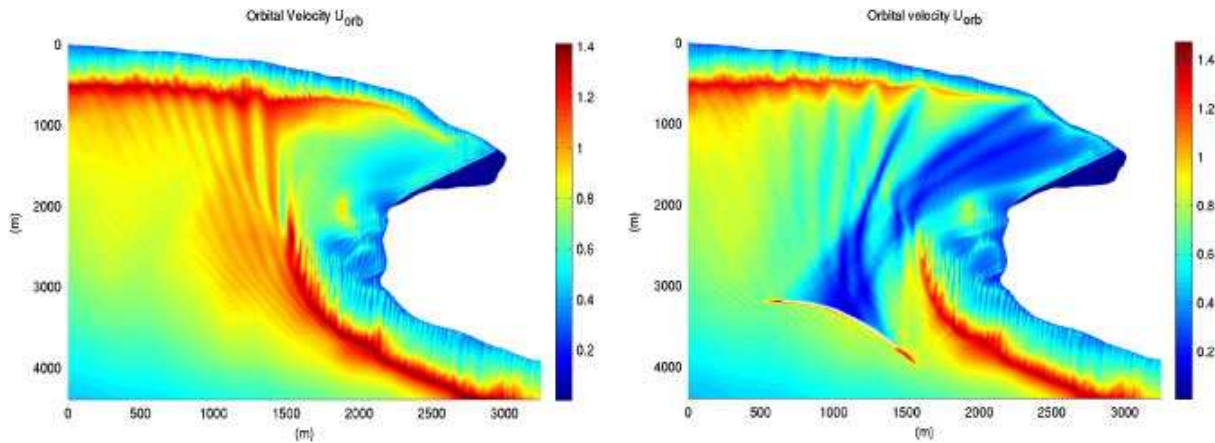


Figure 3.19: The bottom orbital velocity in the domain: (Left) for the initial configuration, (Right) for the optimized configuration.

ure 3.19). This is important to reduce sand suspension and displacement by water waves. Finally, free surface elevation is clearly reduced in region D after geotube introduction (see figure 3.20).

3.2.6 Conclusion

A coastal engineering problem has been modelled and shape design studied for sandy beach protection. Results have shown that geotubes can be designed to reduce beach erosion. And this under the constraint of preserved coastal navigation and minimum water height. These design permits reduction for wave energy, orbital velocity and free surface elevation. Finally, complementary studies have shown that despite not being accounted for during design, the generated currents are also lower after the introduction of the protection device [131].

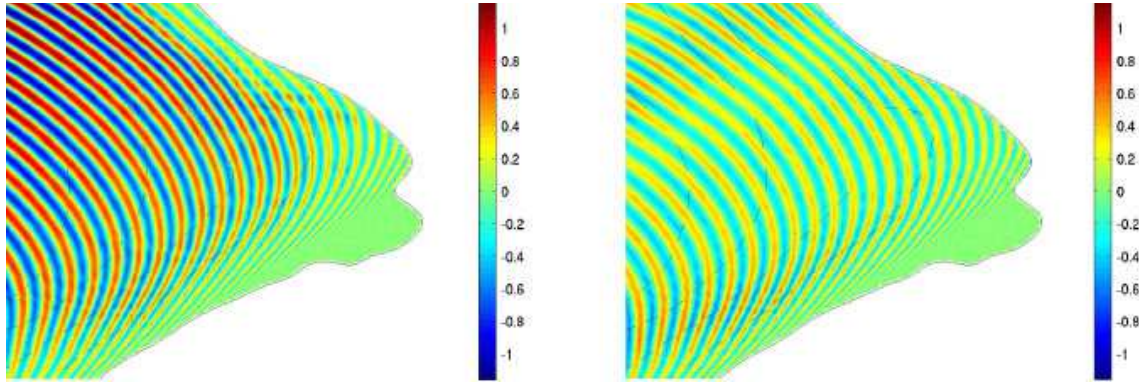


Figure 3.20: The free surface elevation in the cost function region D : (Left) for the initial configuration, (Right) for the optimized configuration.

3.3 Shape optimization of a dispersive Bioreactor

3.3.1 Introduction

The optimal design (i.e., the characterization of the main design parameters of a system under performance/economical/footprint constraints) of biosystems has attracted a lot of attention these last years. Indeed, the diversity of unitary systems and the large spectrum of optimization criteria has led to the search for the optimal solution with respect to a given optimization problem, in particular in the field of catalytic chemical reactions. The problem under consideration in most studies can be stated as follows: given

1. the model of the process,
2. the input and required output reactant concentrations (that is to say, the conversion rate is specified), and
3. the flow rate to be processed,

what are the volumes of N tanks in series such that the total volume of the whole process be minimal?

A rigorous solution to this problem for catalytic biosystems (i.e., a biological reaction in which the activity of the biocatalyst is assumed to be constant) exhibiting Michaelis-Menten kinetics was proposed in [175], while the solution for a fairly large class of autocatalytic systems (including, in particular, the well-known Monod and Haldane kinetics) was proposed by [121]. Recently, these results were revisited and extended by [111], [112] and [209].

However, these studies suffer of two important drawbacks:

- While the proposed results are valid for small and medium sized systems, the diffusion phenomena that occur in larger tanks were not studied;
- The dimensioning parameters were not considered (i.e., only the total volume of the systems were optimized). However, with respect to a real case, design parameters

such as the diameter or the height of any biological or chemical system will influence its performance.

In the present Section, we propose to couple hydrodynamics with biological phenomena occurring in a diffusive bioreactor which the main design parameters (reactor shape and total volume) are optimized with respect to the output concentration. To do so, we present a particular spatial modeling based on coupled PDEs. We also define a second model, computationally cheaper, based on two systems of ODEs with coefficient calibrated using the outputs given by the PDE model. The objective of this second model is to quickly provide the behavior of the considered bioreactor.

This Section is organized as follows. First, in Sections 3.3.2, the PDE and ODE models of the system and the way they are compared are presented. Then, in Section 3.3.3, the optimization problem is defined. Finally, in Section 3.3.4, some preliminary results are given.

3.3.2 Mathematical model

In this Section, we present the two models used to describe the behavior of the considered bioreactors.

Device description

The bioreactor under consideration is depicted in Figure 3.21(a). It contains a certain amount of biomass that resides in Ω^* and reacts with a diluted substrate entering through an inlet Γ_{in}^* that is located at the top. At the bottom there is an outlet Γ_{out}^* allowing the uncontaminated liquid to leave the container.

The device's geometry is that of a solid of revolution and, consequently, it can be characterized by a 2D model. The symmetry axis Γ_{sym} is shown as a dotted line in Figure 3.21(b), the container region is denoted by Ω , the wall is Γ_{wall} , and the inlet and outlet are (respectively) Γ_{in} and Γ_{out} . Note that $\partial\Omega = \Gamma_{\text{in}} \cup \Gamma_{\text{wall}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{sym}}$.

In the numerical experiments that we performed the length of Γ_{wall} was set to 5 m. and the radius of the inlet and outlet were fixed at $\Gamma_{\text{in}} = \Gamma_{\text{out}} = 0.5\text{m}$.

PDEs based modeling

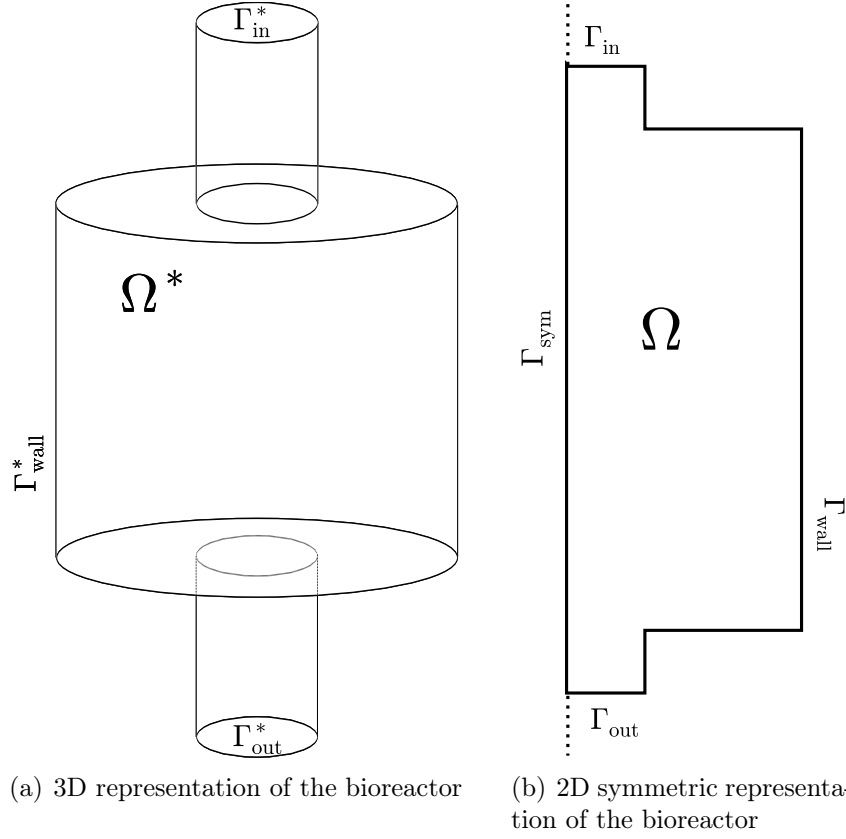
Background material on a similar device can be found in [106].

The fluid is modelled using the Incompressible Navier-Stokes equations

$$\begin{cases} \rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} = \nabla \cdot [-p\mathbf{I} + \eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)] + \mathbf{F}, & \text{in } \Omega \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \end{cases}, \quad (3.3.1)$$

where $\mathbf{u} = (u, v)$ is the velocity field [m/s], p is the pressure [Pa], ρ is the density [kg/m³], η is the dynamic viscosity [Pa · s], and $\mathbf{F} = (0, -g_0 \cdot \eta)$ is the volume force [N/m³], with $g_0 \simeq 9.8$ [m/s²] being the standard gravity constant.

The boundary conditions are: $\mathbf{u} \cdot \mathbf{n} = 0$, in Γ_{sym} , $\mathbf{t} \cdot [-p\mathbf{I} + \eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)] \mathbf{n} = 0$, in Γ_{sym} , $\mathbf{u} = \mathbf{u}_0$, in Γ_{in} , $\eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \mathbf{n} = 0$, in Γ_{out} , $p = 0$, in Γ_{out} , $\mathbf{u} = \mathbf{0}$, in Γ_{wall} , where



$\mathbf{u}_0 = (0, v_0)$ gives a parabolic velocity condition $v_0 = Q(x - \frac{1}{2})(x + \frac{1}{2})$ at the inlet with $Q = 0.2$ m/s.

The process of convection and diffusion of the substrate inside Ω is modelled by:

$$\frac{\partial s}{\partial t} + \nabla \cdot (-D_1 \nabla s) = -\mu(s)x - \mathbf{u} \cdot \nabla s, \text{ in } \Omega, \quad (3.3.2)$$

where s stands for the concentration of the substrate [mol/m³], and $D_1 = 2 \cdot 10^{-8}$ [m²/s] is the diffusion coefficient. The reaction rate μ is a Monod (Michaelis-Menten) kinetic function of the form $\mu(c) = \mu_{\max} \frac{c}{1+c}$, with $\mu_{\max} = 0.05$ [s⁻¹].

The boundary conditions are given by $-\mathbf{n} \cdot \mathbf{N} = N_0$, in Γ_{in} $\mathbf{n} \cdot \mathbf{N} = 0$, in $\Gamma_{\text{sym}} \cup \Gamma_{\text{wall}}$, $\mathbf{n} \cdot (-D_1 \nabla s) = 0$, in Γ_{out} , where $\mathbf{N} = -D_1 \nabla s + s \mathbf{u}$, and the inward flux is given by $N_0 = S_{\text{in}} v$ [mol/(m² · s)] taking $S_{\text{in}} = 1$ in this case. The tank starts with a homogeneous value of x set to 0.5 [mol/m³].

For the biomass, the convection and diffusion are governed by equations similar to (3.3.2) (changing the sign of $\mu(s)x$) with a diffusion coefficient $D_2 = 3 \cdot 10^{-8}$ [m²/s], $N_0 = 0$ [mol/(m² · s)] and an uniform initial bioreactor concentration of 0.5[mol/(m² s)].

This system of PDEs is solved by using a Finite Element Method approach described in [137].

ODE based modeling and comparison with PDE based model

We have performed a comparison between the PDE model and other model based on ODEs systems which model two bioreactors, obtained by dividing the main bioreactor

in two interconnected sub-volumes: one volume is $\alpha \cdot V$, which receive and reject the contaminated flow, and the other one $(1-\alpha) \cdot V$, which is connected with the previous tank. The objective here is to create a model to obtain some relevant bioreactor's behaviors in a quick computational time (the ODE model considered here is less time consuming than the PDE one).

This model is defined by the following dynamical system

$$\begin{cases} \dot{x}_1 = \mu(s_1) x_1 - \frac{\bar{Q}}{\alpha V} x_1 + \frac{d}{\alpha V} (x_2 - x_1), \\ \dot{s}_1 = -\mu(s_1) x_1 + \frac{\bar{Q}}{\alpha V} (S_{\text{in}} - s_1) + \frac{d}{\alpha V} (s_2 - s_1), \\ \dot{x}_2 = \mu(s_2) x_2 + \frac{d}{(1-\alpha)V} (x_1 - x_2), \\ \dot{s}_2 = -\mu(s_2) x_2 + \frac{d}{(1-\alpha)V} (s_1 - s_2), \\ x_1(0) = x_2(0) = S_{\text{in}}, \\ s_1(0) = s_2(0) = 0, \end{cases} \quad (3.3.3)$$

where x_1 and x_2 correspond to the evolution of the biomass and s_1 and s_2 correspond to the evolution of the substrate in the two sub-volumes, $\alpha \in (0, 1)$, $d > 0$, and $\bar{Q} = 4/3\pi r^3 Q$ with $r = 0.5$ being the radius of the inlet and Q having the same value as in Section 3.3.2.

If we denote by $S_{\text{out}}^{\text{ODE}}$ the value of s_1 when [3.3.3] reaches its steady state then we can construct a mapping

$$S_{\text{in}} \mapsto S_{\text{out}}^{\text{ODE}}(S_{\text{in}}; \alpha, d).$$

The equivalent of the previous mapping for **PDE** can be defined as a correspondence between S_{in} and the concentration of substrate at the outlet after enough time has passed:

$$S_{\text{in}} \mapsto S_{\text{out}}^{\text{PDE}}(S_{\text{in}}) = \frac{\int_{\Gamma_{\text{out}}} c \, d\Gamma}{\int_{\Gamma_{\text{out}}} d\Gamma} \quad (3.3.4)$$

These two functions $S_{\text{out}}^{\text{ODE}}$ and $S_{\text{out}}^{\text{PDE}}$ provide us with the basis for comparison of the two models. We performed the following two numerical experiments for the cylindrical shape described in Section 3.3.3 for S_{in} ranging from 0.25 to 10 mol/m³:

ODE1 For each value of S_{in} we computed the values of α and d that minimized

$$(S_{\text{out}}^{\text{ODE}}(S_{\text{in}}; \alpha, d) - S_{\text{out}}^{\text{PDE}}(S_{\text{in}}))^2.$$

The resulting values are shown in Figure 3.3.4.

ODE2 For the set $\{S_{\text{in}}^i \mid i = 1, \dots, 5\}$, we computed the pair α, d that minimized

$$\sum_{i=1}^5 (S_{\text{out}}^{\text{ODE}}(S_{\text{in}}^i; \alpha, d) - S_{\text{out}}^{\text{PDE}}(S_{\text{in}}^i))^2.$$

The resulting values are $\alpha = 0.304$, $d = 0.072$

3.3.3 Optimization problem

Here, we would like to find a shape that results in the most efficient bioreactor. This problem is equivalent to find

$$\arg \min_{s \in \Theta} J(s), \quad (3.3.5)$$

where Θ is the set of all admissible shapes, and J is our fitness function to be defined in the next section. The set of admissible shapes is characterized by the tanks that can be obtained by varying the degrees of freedom labeled with a, b_1, b_2 , and c in Figure 3.21. The contour of Γ_{wall} results from interpolation using a shape-preserving piecewise cubic hermite polynomial.

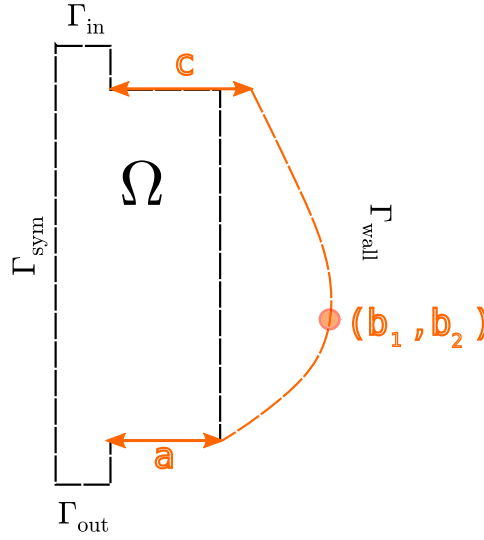


Figure 3.21: Shape parameterization.

Each tank is simulated for approximately half an hour ($t_{\text{max}} = 2000\text{s.}$) and at the end of this period, we compute the flux of the substrate through the outlet using the formula

$$Z = - \int_{\Gamma_{\text{out}}} s v \, d\Gamma.$$

We denote by Z^{cyl} the result of evaluating Z for the cylindrical bioreactor whose shape is characterized by the parameters $a = c = 1.5$, $b_1 = 2.5$, $b_2 = 2$ (its volume is approximately 66 m^3).

The fitness of a given shape is determined according to the expression

$$J = P \max \{0, Z - Z^{\text{cyl}}\} + \text{Volume}, \quad (3.3.6)$$

where P is a penalty taken as 10^9 and Z is the result of evaluating (3.3.2) for the current shape. With this choice of fitness function, the optimization process favors shapes that yield a value of Z smaller than that of the purely cylindrical bioreactor and, among those, it chooses the ones that minimize the tank's volume.

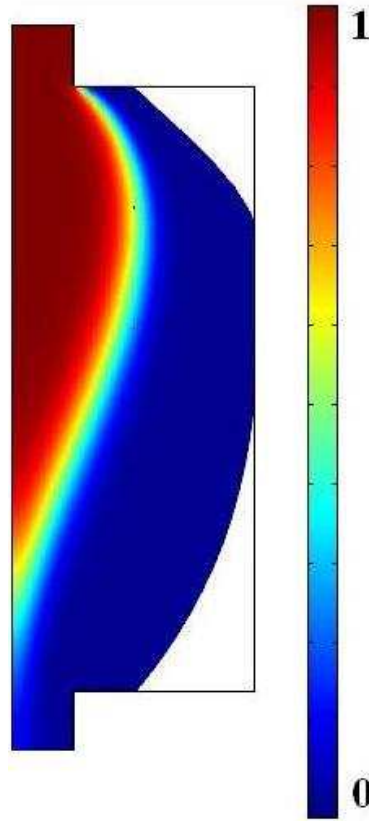


Figure 3.22: Bioreactor shape obtained after the optimization process. The normalized substrate concentration distribution and old shape are also presented.

3.3.4 Results

Optimal shape

In order to solve the optimization problem 3.3.5, we have use the GMSA (with B_1 , $t_{l_1} = 20$, $N_g = 20$, $N_p = 20$, $p_c = 0.45$ and $p_m = 0.35$) described in Chapter 2.

The number of cost function evaluation is about 6000. Each evaluation of the cost function (3.3.6) (implemented using Matlab and COMSOL Multiphysics 3.5a toolboxes [130]), in both 2D and 3D cases, requires about 40s on a Intel Quad-core 2.8Ghz 64bits computer with 12GB of RAM. Thus, the optimization process takes approximatively 67 hours.

After running the optimization procedure described in the preceding section, we obtain the optimal shape displayed in Figure 3.22. The total volume has been reduced by 20% which is a significative improvement of the bioreactor's characteristics.

Comparison between different models

Results are presented in Figure 3.24. As we can observe on this Figure, for each S_{in} there exists a pair (α, d) which fits the PDE and ODE1 models, whose values increase as S_{in} increases. However, the values of d present important oscillations when S_{in} is high.

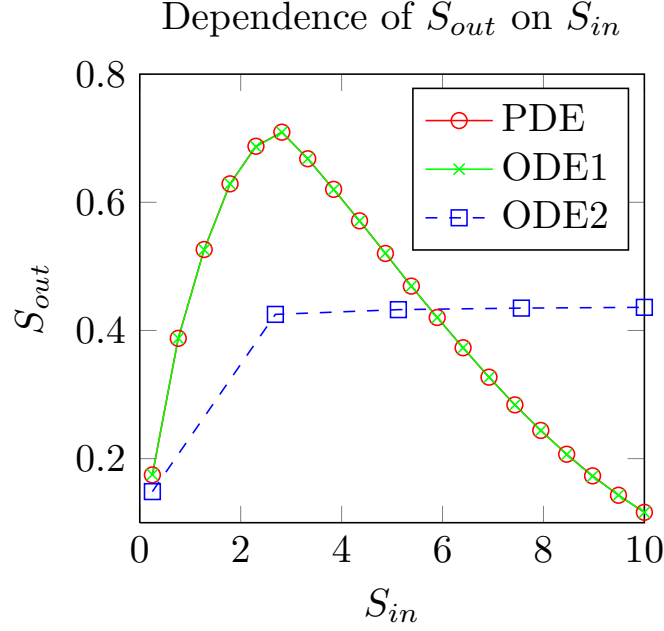


Figure 3.23: Dependence of S_{out} on S_{in} for the three different models: **PDE**, **ODE1** and **ODE2**.

Those oscillations are due to numerical instabilities which can be mitigated by decreasing the time steps used in the ODE model. When solving the multi-objective problem ODE2, we have found values of (α, d) which produce substrate concentrations with tendencies similar to the PDE model (increasing then constant). A more in-depth analysis should be conducted in order to better match the ODE2 and PDE models. For instance, this could be achieved by increasing the number of ODEs and variables considered.

3.3.5 Conclusions and Perspectives

In this Section, we have presented two models, based respectively on PDEs and ODEs, for describing the behavior of a particular bioreactor. The PDE model has been used to formulate an optimization problem which consisted in reducing the bioreactor volume keeping its cleaning efficiency. It was solved with the GMSA and results show a volume reduction of 20%. The ODE model has been calibrated to reproduce similar results than the PDE model in a lower computational time.

Those first results are encouraging and further studies should contemplate a more complex shape optimization (for instance, considering the bioreactor height) and a more complete ODE model (involving more ODEs and coefficients). Among possible extensions, we should consider cases where the bioreactor is equipped with a system of retention of biomass (either moving or fixed bed bioreactor).

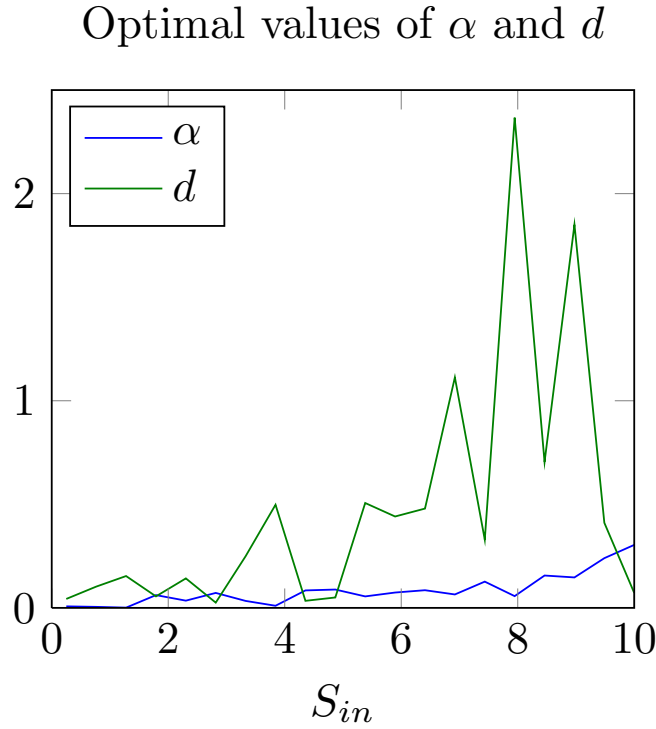


Figure 3.24: Optimal values of α (continuous line) and d (dotted line) for each value of S_{in} in **ODE1**.

3.4 Design of Code Division Multiple Access Filters Based on Sampled Fiber Bragg Grating by Using a Shape Optimization Approach

3.4.1 Introduction

The use of optical fibers offering huge bandwidth in the telecommunication sector has known important developments in the last decade [248, 274] with applications such as cable television or cell phones [254, 273]. The advantages of optical fibers are numerous as, for instance, they offer: large communication capacity, low transmission loss, low cost and easy hybridization with other optical devices [277]. However, to be fully efficient, the fibers should allow multiple access. This means allowing various persons (called **users**) to send and receive messages in the fiber at the same time (See Figure 3.25).

There exist three main schemes of multiple access: Time Division Multiple Access (TDMA) [160] which allows to consider a great number of users but requires fast synchronization, Wavelength-Division Multiplexing (WDM) [29] which sometimes requires precise adjustments, and Code Division Multiple Access (**CDMA**) [266] which primarily allows a great flexibility in multiple accesses. This last technique, which consists in allocating individualized codes to represent the messages of each user, exhibits various advantages such as high fidelity and high resistance to signal perturbations, secured communications

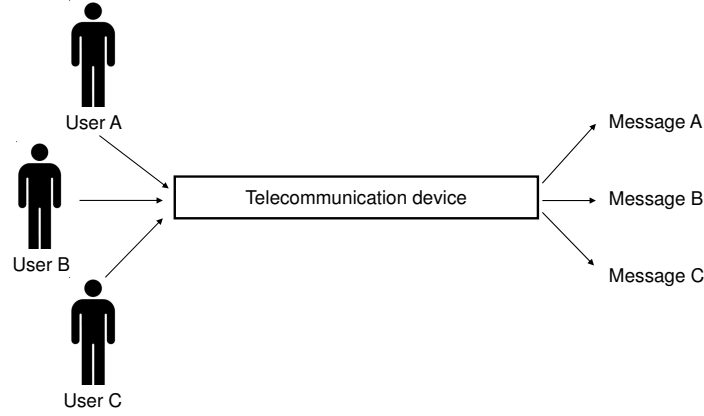


Figure 3.25: Multiple access technique principle: Users A,B and C send at the same time a message in a communication device. The multiple access technique allows to properly deliver each message.

and low power consumption [145].

Focusing on CDMA, an efficient and low-cost way to implement this approach is to consider an optical spectral representation of the user codes [150, 161]. This implementation is called optical CDMA (OCDMA). Those codes are then generated by optical guides, and in particular by sampled fiber Bragg gratings (**SFBG**) [81, 277]. SFBGs are optical fibers based on a periodic perturbation of their refractive effective index, obtained by exposing it to UV radiations [49, 77, 179]. The objective of SFBGs is to reflect predetermined wavelengths and to let other wavelengths pass [48, 269]. In a OCDMA device we are interested in designing SFBGs that reflect part of the wavelengths corresponding to the code representing a particular user. This objective can be mathematically reformulated as an inverse problem that consists in determining the effective refractive index profile (i.e., shape) of a SFBF knowing a priori its reflected spectrum. To solve this inverse problem we consider global optimization algorithms [248]. Currently, one of the most used optimization methods in this domain are genetic algorithms (GA) [80, 46, 256].

In this Section, we focus on the design of a CDMA filter, composed of a SFBF, used to reflect the whole spectral CDMA code of a user. As previously said, we could solve this problem by considering a GA. However, we would like to see if the target can be achieved with a lower computational effort by using the SDMSA-L2 and the GMSA presented in Chapter 2 [134, 136, 138]. In order to check the efficiency of our algorithms on the considered SFBG design problem, we compare their results with those given by a traditional GA.

The work is organized as follows. Section 3.4.2 briefly describes the CDMA filter based on SFBG, its mathematical modeling and the associated design problem. In Section 3.4.3, we consider a particular case and compare the results obtained by the different

optimization algorithms in term of precision and computational time.

3.4.2 CDMA filter design problem

In this Section, we first introduce some basic principles of CDMA and the SFBG to be designed. Then, we recall some previous works on SFBG optimization. Next, we present the mathematical model that allows to compute the reflected spectrum of a SFBF and we reformulate the design problem as an inverse problem.

CDMA principle and implementation

In the basic technique of CDMA, the bits '1' or '0' of a binary message, send by a user, are replaced at the level of the transmitter by two codes attributed to this user [266]. For instance, we can consider CDMA binary codes of length $N_{\text{code}} \in \mathbb{N}$. The code for the bit '1' of a particular user 'A' is denoted by $c_1^A \in \{0, 1\}^{N_{\text{code}}}$ and its complement, denoted by $c_0^A = -(c_1^A - 1)$, is used for '0' (e.g., if $N_{\text{code}} = 8$, two possible codes are $c_1^A = '10110011'$ and $c_0^A = '01001100'$). During this work, we only focus on this binary CDMA coding method used, for example, when considering the so-called Gold codes [75].

In order to implement this approach, Kavehrad et al. [150] and later Lam [161] suggested to use spectra composed of a set of wavelengths $\Lambda = (\lambda_i)_{i=1}^{N_{\text{code}}}$ to transmit CDMA binary codes of length N_{code} through optical devices (such as optical fibers). In that case, we represent a particular CDMA code c by a spectrum Λ_c whose reflectivity at wavelength λ_i is equal to $c(i)$, for $i = 1, \dots, N_{\text{code}}$, and zero elsewhere. For instance, assuming $N_{\text{code}} = 5$, the spectrum showed in Figure 3.26-(Left) represents the code '10101' whereas the one depicted by Figure 3.26-(Right) corresponds to the code '11010'.

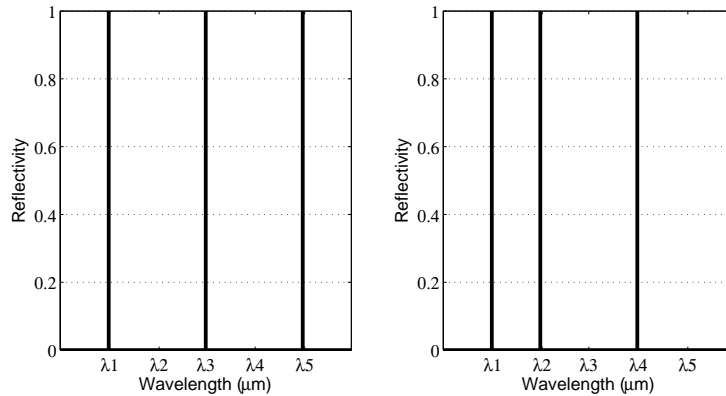


Figure 3.26: Example of spectra representing the CDMA binary codes '10101' (**Left**) and '11010' (**Right**), when $N_{\text{code}} = 5$.

One way to generate such a spectrum is to consider CDMA filters composed of SFBGs [81, 277]. As said in Section 3.4.1, SFBGs are optical fibers with a periodic perturbed effective refractive index that reflect predetermined wavelengths and let other wavelengths pass. When applied to CDMA filters, their objective (obtained by combining the effect

of various SFBGs) is to reflect the spectrum Λ_c corresponding to the CDMA code c of a particular user.

In this work, we are interested in designing, by considering an optimization approach, a SFBF that reflects the whole spectrum associated to a CDMA code. The advantages of such an approach in comparison to other ones (for instance, combining various SFBGs) is the reduction of time and cost of CDMA filter fabrication [224].

Previous works on SFBG optimization

Nowadays, the range of industrial applications of SFBGs is wide [120]. For instance, they can be used as: dispersion compensation [270], sensors [151], network monitoring [44] or wavelength selective devices (as in this Section).

Considering this last domain, the SFBG can be employed as a pass-band filter [256] or a multichannel filter [48, 269]. There is therefore an important demand for optimization methods generating SFBGs giving a desired reflectivity spectrum [241]. Furthermore, those method need to perform global optimization as it is observed that the considered functionals usually have multiple minima [248].

In the literature, various optimization algorithms have been applied to solve such problems, considering some SFBG characteristics (explained in Section 3.4.2) as the optimization parameters. For example, Navruz et al. use a Simulated Annealing approach to optimize the phase profile of a Multichannel SFBG [207], whereas Ngo et al. consider a Tabu Search algorithm to design the index modulation function of a pass-band SFBG [211]. However, one of the most used method to solve SFBG design problems are GAs as in the works presented in [46, 80, 248] and [256], where various SFBG parameters (such as the fiber length or the refractive index modulation length) are optimized to obtain pass-band filters.

Regarding this previous literature, the optimization algorithms are only presented as a tool for SFBG design without deeper details (e.g., the algorithm parameters, the number of cost function evaluations or the numerical precision of the solution are not given). Moreover, in each article the optimization parameters and the cost function are different. Thus, comparing the results and efficiency of an optimization approach with the ones given in those articles is difficult. For this reason, when we started to study, in collaboration with the company “Alcatel” (<http://www.alcatel-lucent.com>) and the “Département Photoniques et Ondes” of the Université Montpellier 2 (<http://www.ies.univ-montp2.fr>), SFBG design problems by using our own optimization method (presented in Chapter 2), we decided to compare the obtained results with those provided by our own GA (as it is the most used method in this domain). We point out that we have previously applied this methodology to the design of optical fibers simpler (regarding the fiber structure and reflected spectra) than the ones considered in this work [134, 135]. We obtained better results with our algorithms than with those given by GAs (in terms of number of functional evaluations and result precision). The work presented here is a continuation of that previous paper.

SFBG reflected spectrum evaluation model

In order to design the SFBG described in Section 3.4.2, we need to introduce the mathematical model that allows us to compute its reflected spectrum.

We consider a SFBF of length L (mm). We assume that for any wavelength λ (μm), in a considered transmission band $[\lambda_{\min}, \lambda_{\max}]$, sent in the SFBG, part of this wavelength is transmitted through the fiber with a varying amplitude $T(z, \lambda)$ at the position $z \in [0, L]$ in the fiber and other part is reflected with a varying amplitude $R(z, \lambda)$. Furthermore, for each $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ and each $z \in [0, L]$, we have that $T(z, \lambda)$ and $R(z, \lambda)$ are coupled through the following equations [77, 135]:

$$\begin{cases} \frac{dT}{dz}(z, \lambda) = i(2\pi \frac{n_{\text{eff}} + \delta n_{\text{eff}}(z)}{\lambda} - \frac{\pi}{\Theta})T(z, \lambda) + i\frac{\pi \delta n_{\text{eff}}(z)}{\lambda}R(z, \lambda), \\ \frac{dR}{dz}(z, \lambda) = -i(2\pi \frac{n_{\text{eff}} + \delta n_{\text{eff}}(z)}{\lambda} - \frac{\pi}{\Theta})R(z, \lambda) - i\frac{\pi \delta n_{\text{eff}}(z)}{\lambda}T(z, \lambda), \\ T(0, \lambda) = 1, \\ R(L, \lambda) = 0, \end{cases} \quad (3.4.1)$$

where n_{eff} is the unperturbed effective refractive index; Θ is the nominal grating period (μm); and $\delta n_{\text{eff}}(z)$ is the slowly varying index amplitude change over the grating (also called, in the following, **apodization**) which is periodic.

System (3.4.1) is solved numerically by using a simplified transfer matrix method [18].

Finally, the **reflected spectrum**, denoted by r , of the considered SFBG is defined as:

$$r(\lambda) = |R(0, \lambda)|^2. \quad (3.4.2)$$

Inverse problem formulation

We consider that the code c_1^A , introduced in Section 3.4.2, is represented by the following set of wavelengths:

$$\Lambda_A^1 = \{\lambda_i | i = 1 \dots N, N \leq N_{\text{code}}, \lambda_i \in \Lambda\}. \quad (3.4.3)$$

Due to the fact that the considered SFBG introduced in Section 3.4.2 can only generate symmetric reflected spectra [96], we want to design a SFBF that reflects Λ_A^1 and also the symmetrical wavelengths of Λ_A^1 centered around a predefined wavelength λ_c . This new set of wavelengths is denoted by Λ_{A, λ_c}^1 .

This problem can be reformulated considering that each SFBG can be characterized by its apodization profile $\delta n_{\text{eff}}(z)$. By denoting Ω_{apo} the search space of all admissible apodization profiles, we define a cost function h_0 , to be minimized on Ω_{apo} , by:

$$h_0(x) = \int_{[\lambda_{\min}, \lambda_{\max}]} (r_x(\lambda) - r_t(\lambda))^2 d\lambda, \quad (3.4.4)$$

where $r_x(\cdot)$ is the reflected spectrum (3.4.2) of the SFBG with an apodization profile associated to $x \in \Omega_{\text{apo}}$ and r_t is the target reflected spectrum given by:

$$r_t(\lambda) = \begin{cases} 1 & \text{if } \lambda \in \Lambda_{A, \lambda_c}^1, \\ 0 & \text{elsewhere.} \end{cases} \quad (3.4.5)$$

We must include some restrictions on Ω_{apo} in order to find a SFBF with an apodization profile with suitable characteristics for practical realization. Indeed, complex profiles would require high-level and expensive mastering of the writing process [49]. In particular, we are interested by admissible profiles which have a low number of π -phase shifts (sign changes), are smooth and have a maximum index variation \bar{n}_{max} of less than 5.10^{-4} [224]. Thus, apodization profiles are generated by spline interpolation through a reduced number of N_S points equally distributed along the first half of their periodic pattern and completed by parity and periodicity [135]. N_S is chosen high enough to ensure enough peaks in the reflected spectra but small enough for the profile to remain admissible [134].

Thus, the corresponding search space of the optimization problem is:

$$\Omega_{N_S} = [-\bar{n}_{\text{max}}, \bar{n}_{\text{max}}]^{N_S}. \quad (3.4.6)$$

The discrete version of the cost function (3.4.4) on Ω_{N_S} is defined by:

$$h_{0,N_c}(x) = \sum_{i=1}^{N_c-1} \frac{(\lambda_{i+1} - \lambda_i)}{2} \left[(r_x(\lambda_{i+1}) - r_t(\lambda_{i+1}))^2 + (r_x(\lambda_i) - r_t(\lambda_i))^2 \right]. \quad (3.4.7)$$

In Equation (3.4.7), the reflected spectrum r_x of the SFBG with an apodization profile associated to $x \in \Omega_{N_S}$ is evaluated on N_c wavelengths equally distributed on the transmission band $[\lambda_{\min}, \lambda_{\max}]$.

Therefore, the SFBG design problem can be reformulated as the following optimization problem:

$$\begin{cases} \text{Find } x_m \in \Omega_{N_S} \text{ such that} \\ h_{0,N_c}(x_m) = \min_{x \in \Omega_{N_S}} h_{0,N_c}(x). \end{cases} \quad (3.4.8)$$

3.4.3 Numerical test

In this Section, in order to illustrate the efficiency (i.e., computational time and result precision) of our optimization methods, described in Chapter 2, to solve the SFBG design in comparison to the GA approach, we apply the SDMSA-L2 and GMSA to solve a particular implementation of Problem (3.4.8) based on a realistic case. First, in Sections 3.4.3 and 3.4.3, we introduce the parameters associated to the considered SFBG and optimization algorithms, respectively. Finally, in Section 3.4.3 we present and analyze the obtained results.

Here, we present the solutions found for one particular numerical case and we consider that they are representative of the general behavior of our algorithms on this kind of problem. Indeed, other numerical examples, presented in part in [134], have produced similar results.

SFBG characteristics

We consider a CDMA binary codification of length $N_{\text{code}} = 8$ represented by the set of wavelengths $\Lambda = \{ \lambda_1 = 1.5465\mu\text{m}, \lambda_2 = 1.5473\mu\text{m}, \lambda_3 = 1.5481\mu\text{m}, \lambda_4 = 1.5489\mu\text{m}, \lambda_5 = 1.5497\mu\text{m}, \lambda_6 = 1.5505\mu\text{m}, \lambda_7 = 1.5513\mu\text{m}, \lambda_8 = 1.5521\mu\text{m} \}$.

In this case, the particular code $c_1^A = '10110011'$ is characterized by $\Lambda_A^1 = [\lambda_1, \lambda_3, \lambda_4, \lambda_7, \lambda_8]$. We are interested in designing that reflects Λ_{A,λ_c}^1 with $\lambda_c = 1.5525\mu m$. This spectrum is depicted by Figure 3.27.

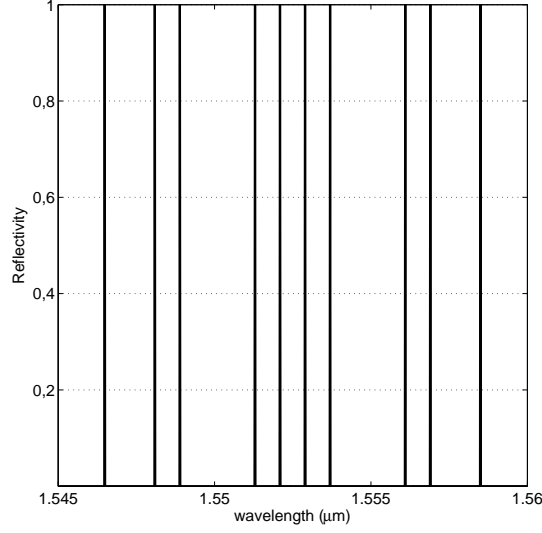


Figure 3.27: Target reflected spectrum of the SFBG presented in Section 3.4.3.

The SFBG characteristics are set to $n_{\text{eff}} = 1.45$, $L = 100\text{mm}$, $\Theta = 0.53\mu m$ and the period of δ_{neff} is 1.039mm . Those values correspond to real optical fiber data provided by our collaborators from “Alcatel” and the “Département Photoniques et Ondes” [224]. Furthermore, the SFBG apodization profiles are generated by $N_S = 9$ interpolation points with $\bar{n}_{\text{max}} = 5.10^{-4}$ and the functional (3.4.7) is evaluated considering $N_c = 1200$ wavelengths in the transmission band $[1.545\mu m, 1.56\mu m]$. This set of parameters give good results (apodization profiles easy to implement) considering the problem of designing a multichannel optical filter of 16 peaks [135].

Parameters of the optimization algorithms

In order to solve problem (3.4.8), considering the SFBG characteristics given in Section 3.4.3, we use the GA, the SDMSA-L2 and the GMSA (see Chapter 2) with the following parameters:

- For SDMSA-L2: We use a two-layer algorithm A_2 with $t_0 = 10$, $t_{l_1} = 5$, $t_{l_2} = 5$ and $\epsilon = 0$. The initial point v_1 for A_2 is generated randomly in Ω_{N_S} . The gradient of h_{0,N_c} used in A_0 is approximated considering a finite difference method.
- For GMSA: We consider B_1 , $t_{l_1} = 50$, $N_g = 10$, $N_p = 10$, $p_c = 0.45$ and $p_m = 0.35$.
- For GA: We use the same stochastic processes than GMSA but with a different set of parameters: $N_g = 30$, $N_p = 180$, $p_c = 0.35$, $p_m = 0.15$. 10 iterations of

the steepest descent method are performed at the end of the GA starting from the obtained solution.

SDMSA-L2, GMSA and GA applied with those sets of parameters have been validated on various benchmark test cases [134, 138] and industrial applications [62, 43, 4, 132, 137, 136], in particular on the design of pass-band and multichannel optical filters [135].

Results and discussion

Figure 3.28 shows the apodization profiles obtained with SDMSA-L2, GMSA and GA and Figure 3.29 shows their associated reflected spectra. The convergence histories of each optimization process are presented in Figure 3.30. Results reported in this Section are summarized in Table 3.9.

For SDMSA-L2, the final value of the cost function h_{0,N_c} is equal to 2.09. The total number of functional evaluations is about 3000. SDMSA-L2 optimization takes approximately 10 hours real time in a 3.4GHz PC computer with 1 Gb Memory.

For GMSA, the final cost function h_{0,N_c} is equal to 2.31. The total number of functional evaluations is about 2700. GMSA optimization takes 9 hours.

For GA, the final cost function h_{0,N_c} is equal to 2.38. The total number of functional evaluations is about 5600. GA optimization takes 18 hours 40 minutes.

The three optimized apodization profiles have different shapes and are situated in distinct attraction basins of the function h_{0,N_c} . This points out the fact that h_{0,N_c} is highly non convex and the optimization problem (3.4.8) difficult to solve. This is confirmed by observing the convergence history of the SDMSA-L2, which shows that, during the optimization process, the steepest descent algorithm has visited various attraction basins and found different local minima.

From a numerical points of view, both SDMSA-L2 and GMSA have found better results than GA and are less time consuming.

From an implementation point of view, all optimized apodization profiles present interesting characteristics:

- As we can observe on Figures 3.27 and 3.29, the reflected spectra associated to the optimized profiles correspond to good approximations of the target reflected spectrum.
- The optimized apodization profiles (see Figure 3.28) are suitable for practical implementation. Indeed, the number of necessary π -phase shifts is 5 (a number easy to implement), the index modulation of the profile is uniformly distributed along the pattern and the maximum index variation of the profile is inferior to $3 \cdot 10^{-4}$, which is a reasonable level [135, 224].
- A stability analysis on the reflected spectra, when applying a random perturbation of 10% on the optimized apodization profiles, show that all optimized results have a small variation of $\approx 4.3\%$ on their reflected spectrum (considering a L^2 norm). This is important because, due to technical limitations, small perturbations could appear in the apodization profile during the writing process.

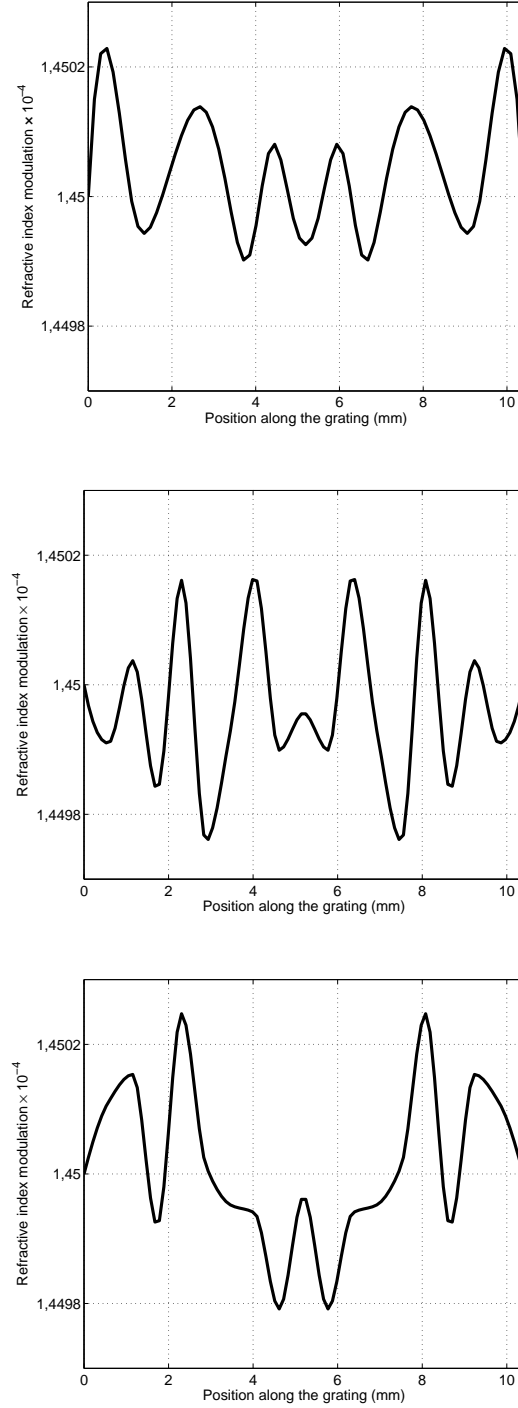


Figure 3.28: Optimized apodization profiles obtained by SDMSA-L2 (**Up**), GMSA (**Middle**) and GA (**Bottom**) optimization methods when solving the SFBG design problem presented in Section 3.4.3.

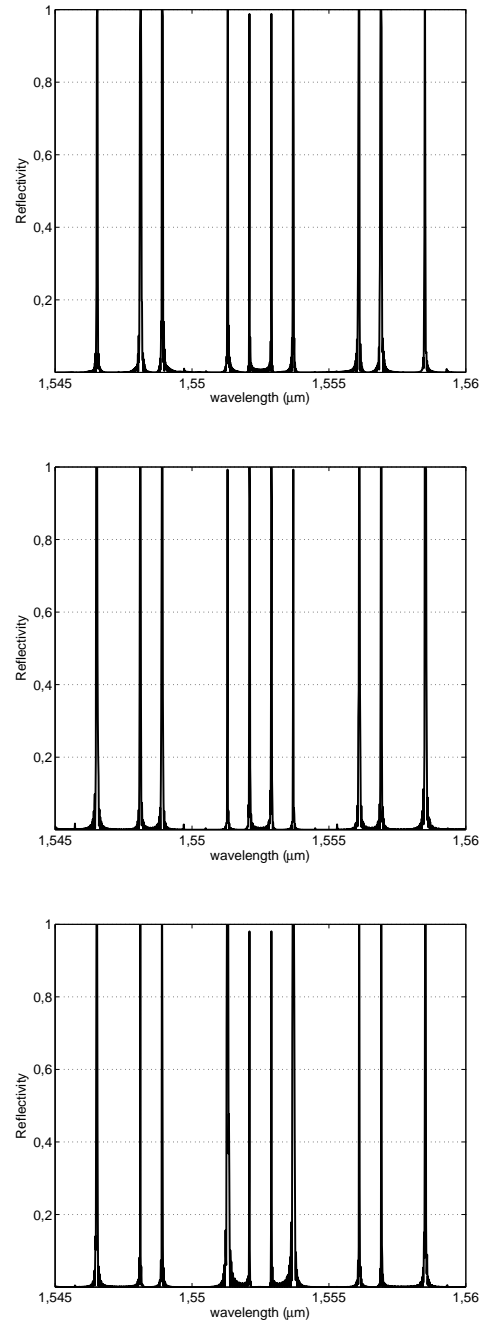


Figure 3.29: Reflected spectrum associated with the optimized apodization profiles obtained by the SDMSA-L2 (**Up**), GMSA (**Middle**) and GA (**Bottom**) optimization methods when solving the SFBG design problem presented in Section 3.4.3.

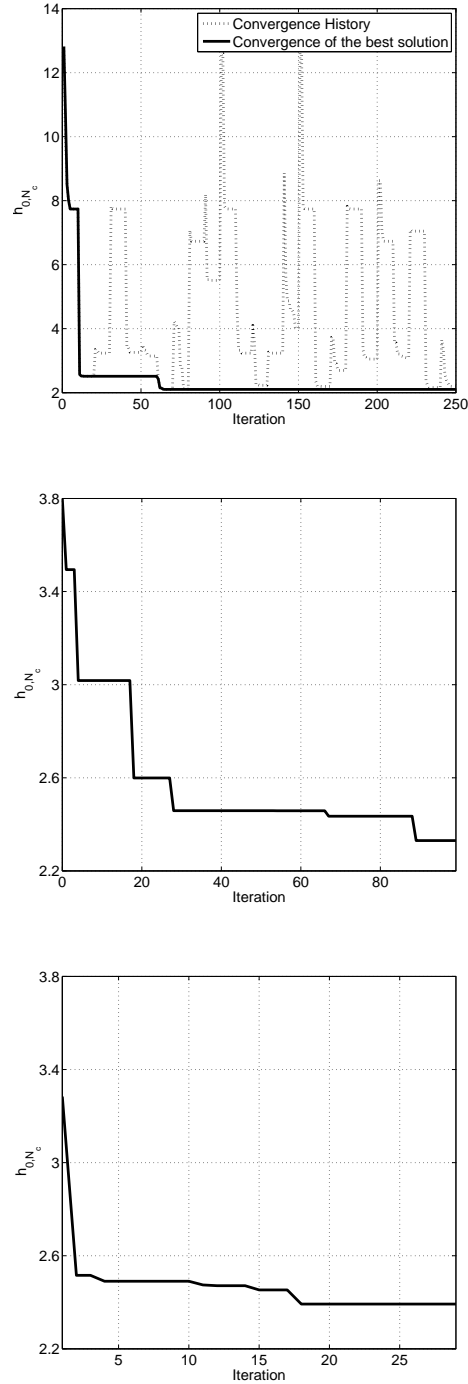


Figure 3.30: Convergence histories obtained when solving the SFBG design problem presented in Section 3.4.3: (**Up**) best element convergence history vs. iterations (—) and global convergence history vs. iterations (...) for SDMSA-L2; (**Middle**) best element convergence history vs. iterations for GMSA; and (**Bottom**) GA .

Optimization Method	Final Cost Function Value	Evaluation Number
SDMSA-L2	2.09	3000
GMSA	2.31	2700
GA	2.37	5600

Table 3.9: Numerical results obtained considering (**Left**) SDMSA-L2, GMSA and GA optimization methods, (**Center**) value of h_{0,N_c} of the best element found by the optimization algorithm, (**Right**) Number of evaluation of the function h_{0,N_c} needed by the optimization algorithm.

3.4.4 Conclusions and Perspectives

A particular code division multiple access filter based on sampled fiber Bragg grating has been designed by using three particular optimization algorithms: two original MSA (SDMSA-L2 and GMSA) and a genetic algorithm (GA) (considered as a classical method to design SFBG). The apodization profiles produced by those optimization approaches exhibit good characteristics for practical implementation because they have no step variation, low maximum index modulation values and small numbers of π -phase shifts. Also, their associated reflected spectrum are weakly sensitive to perturbations. However, SDMSA-L2 and GMSA have produced better solutions (in term of cost function value) and need less computational time (approximatively a half) than GA alone.

A next step, could be the study of the effect of combined apodization and phase profiles optimization [241] in order to avoid the symmetry in spectra mentioned previously. In fact, during this work, we have been interested only by apodization optimization to keep a grating easy to implement by any optical laboratory [49, 179]. Indeed, phase variation requires more complex and expansive materials.

Chapter 4

An Epidemiological model for studying Classical Swine Fever Virus spread between and within farms.

Abstract - *Classical Swine Fever is a viral disease of pigs that causes severe restrictions on the movement of pigs and pig products in the affected areas. The knowledge of its spread patterns and risk factors would help to implement specific measures for controlling future outbreaks. In this Chapter, we describe in detail a spatial hybrid model, called Be-FAST, based on the combination of a stochastic Individual-Based model (modeling the interactions between the farms, considered as individuals) for between-farm spread with a Susceptible-Infected model for within-farm spread, to simulate the spread of this disease and identify risk zones in a given region. First, we focus on the mathematical formulation of each component of the model. Then, in order to validate Be-FAST, we perform various numerical experiments considering the Spanish province of Segovia. Obtained results are compared with the ones given by two other Individual-Based models and real outbreaks data from Segovia and The Netherlands. Finally, various internal validity and sensitivity analysis experiments are used to evaluate the overall reliability and robustness of the model outputs.*

4.1 Introduction

Modeling and simulation are important tools to fight diseases [8]. Each disease has its own characteristics and, therefore, most of them need a well-adapted simulation model in order to tackle real-life situations [32].

In this Chapter, we consider Classical Swine Fever (CSF). CSF is a highly contagious viral disease of domestic and wild pigs caused by the CSF Virus (CSFV) [199]. It generates important economic losses (due to, for instance, the devaluation of the pig meat price, the cost of culling and cleaning infected farms, the price of detection campaigns, etc.) in the affected regions [83, 212]. Despite the efforts to control and eradicate CSF, it remains endemic in many countries of America, Africa and Asia and sporadic outbreaks have affected 19 European countries from 1996 to 2007 (See Table 4.1 and [74]). Due to the

Country	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007
Serbia	73	62	69	172	128	52	44	110	489	401	
Ukraine					2						
Switzerland		47	26								
Russia	10	11	14	16	14	9	6	4	8	2	7
Romania					1	52	155	182	1075	1338	159
U.K.				16							
Netherlands	424	5									
Luxembourg					7	77	5				
Italy	55	18	9	3	5			1			
France						9	13	7	1		1
Spain	78	21			33	16					
Slovakia	97	90	14	21	40	48	37	12	4	5	
Czech Rep.	2	21	13								
Croatia	10		2						7	42	2
Bulgaria	8	8	4	1	1	32	11	4	5	2	1
Bosnia-Herz.	43		5	1	52	90	84	23	40	35	33
Belgium	8					1					
Austria					1						
Germany	44	11	415	174	378	451	38	3	24	52	11

Table 4.1: CSF outbreaks in Europe 1997-2007 [74].

different ways of CSFV spread (airborne, contact with infected animals, etc.) [76, 236], it is difficult to extrapolate the routes of infection and the consequences of a CSF epidemic from one region to another. Furthermore, the magnitude and duration of a CSF epidemic change depending on the epidemiological and demographic characteristics of the infected area and the timing and effectiveness of the applied control measures [180].

The study of the potential spread patterns of CSFV into a region may help to identify risk zones to improve the prevention and management of future outbreaks. In CSF-free areas, a good way to quantify the magnitude of potential CSF epidemics and evaluate the efficiency of control measures is to use simulation models. Recently, some models have been developed to simulate CSFV spread into CSF-free regions such as Belgium, Germany, Australia and The Netherlands [140, 147, 197, 252]. In [187], authors have also described a spatial stochastic model for Spain by using the commercial software InterSpread Plus [244, 188]. However, most of those models only focus on the between-farm spread of the CSFV, with poor assumptions regarding the within-farm spread and do not explicitly consider the specific farm to farm contact patterns (such as commercial network, shared vehicles, etc.) into the studied region.

To overcome these limitations, a spatial hybrid model, called Be-FAST (Between Farm Animal Spatial Transmission), has been developed [185, 186]. Compared to the other models introduced previously, Be-FAST presents interesting novel characteristics. One of the most important is that it is based on the combination of a stochastic Individual-Based model (where the farms are considered as individuals) [61, 147], simulating the between-farm interactions (such as pig shipments, veterinarian routes, etc.) and CSFV spread, with

a Susceptible-Infected model [32], simulating the within-farm CSFV spread. In particular, Be-FAST uses the proportions of infected pigs in each contaminated farm to calibrate dynamically some of its coefficients (for instance, those referring to the probabilities of infection). Another newsworthy feature of Be-FAST is the use of real database for farms and transports allowing simulating realistic commercial contacts between farms. We note that, in recent literature, other epidemiological models consider a similar approach (for instance, in [177], the authors introduce a model based on a Monte-Carlo algorithm which uses real farms and transports databases but does not simulate the number of infected animals within each farm). At the end of a simulation, Be-FAST returns outputs referring to CSF outbreaks characteristics (such as the mean epidemic magnitude, the CSFV introduction risk map, the proportions of CSF infection routes and the proportions of detection due to control measures).

This Chapter has three objectives. The first one is to give a complete mathematical formulation of Be-FAST in order to provide a transparent and understandable model for users. The second one is to validate and check the robustness our model. To do so, we compare some of the Be-FAST results obtained considering two particular numerical experiments with the outputs given by two other models (InterSpread Plus [244] and the Individual-Based model presented in [147]) and real outbreak information. Those experiments simulate the possible evolution of a CSF epidemic inside the particular Spanish province of Segovia [143]. In the first case, the CSF epidemic spreads freely without any control measure whereas in the second case we consider all the control measures defined by the Spanish administration. Similar validation approach has been considered in previous literature (for instance, see [140, 149]). Then, in order to check the robustness of our model, we perform various internal validity (i.e., consistency of the results after several runs of a stochastic model) and sensitivity analysis experiments [243].

Here, after recalling in Section 4.2 the main characteristics of the CSF, we give an extended description of the Be-FAST model from the mathematical perspective (i.e., detailed equations, numerical schemes, etc.) in Section 4.3. Next, during Section 4.4, we focus on the model validation by considering various numerical experiments based on real databases provided by the Spanish administration [143, 182]. We also compare the results given by our model with those obtained by other models [244, 149] and data observed during CSF outbreaks occurring in Spain and The Netherlands [184, 76, 140]. Finally, in Section 4.5, we describe the internal validity and sensitivity analysis processes and analyze the robustness of the model.

4.2 Characteristics of Classical Swine Fever

In order to facilitate the understanding of the Be-FAST model, described in Section 4.3, we briefly explain the CSF evolution process, the routes of CSFV transmission and some control measures used to fight CSFV. A complete justification of the assumptions and simplifications considered during this work can be found in [185].

4.2.1 CSF evolution

CSF results from infection by CSFV, a member of the genus Pestivirus and family Flaviviridae [199]. CSFV affects both domestic and wild pigs. When a pig is not infected by CSFV, it is categorized in the Susceptible state. Once it is infected, it passes successively through the following states [76]:

- **Infected:** The pig is infected by CSFV but cannot infect other pigs and has no visible clinical signs (i.e., fever, lesions, etc.). The mean duration of a pig in this state is 7 days and is called latent period. Then, the pig passes to be infectious.
- **Infectious:** The pig can infect other pigs but does not have clinical signs. The mean duration from infectious to the development of clinical signs is 21 days and is called incubation period. After that period, the pig presents clinical signs.
- **Clinical Signs:** The pig develops visible clinical signs and still may infect other pigs. After a period between two weeks and three months the pig can be recovered or died due to the disease. However, the CSF death and recuperation of pigs are assumed to be neglected because the culling of infected animals is usually accomplished before.

Those four states can be also applied at the farm level by considering that a farm is in the state (in order of priority) [147]:

- **Clinical Signs (denoted by CF):** If at least one pig has clinical signs.
- **Infectious (denoted by TF):** If at least one pig is infectious.
- **Infected (denoted by IF):** If at least one pig is infected.
- **Susceptible (denoted by SF):** If all pigs in the farm are susceptible.

A farm either in the state IF , TF or CF is called contaminated farm. Moreover, a farm in the state TF or CF is also called spreading farm.

4.2.2 Routes of transmission

The main ways of CSFV transmission (i.e., that a susceptible pig becomes infected) are: the direct contacts with infected animals; and the indirect contacts due to airborne spread of the CSFV and contaminated vehicles, persons or fomites (i.e., material). Historically, those ways of spreading have been reported as the principal routes of CSFV spread [76, 236], although other routes (such as movements of wild animals) have also been described as potential ways of CSFV transmission but with a minor impact on the CSF epidemics [76]. Those alternative routes have been neglected here.

4.2.3 Control measures

Once an animal becomes infected, another important concept in epidemiology is its detection and the application of control measures by the authorities [199].

When an infected pig is detected in a farm, this farm is classified as detected. Generally, before the first detection of a contaminated farm (called index case), the detection occurs when pigs present clinical signs and is due to the vigilance of the farmers or veterinarians [158]. After the detection of the index case, the awareness of the farmers and authorities is widely increased and the time of detection decreases [147]. Moreover, the detection can be also due to the control measures presented below.

In order to control a potential CSF epidemic, the following control measures defined by the European and Spanish legislations, described in [140, 143, 184, 182], are considered:

- Movement restrictions: Outgoing or incoming movements in farms inside the considered region are limited during a specified time period (in our case, between 1 and 3 months).
- Zoning: Zones, called protection and surveillance zones, are defined around a detected farm (considering a minimum radius of 3 km and 10 km, respectively). Surveillance activities are applied within those zones during 30 and 40 days, respectively.
- Depopulation: All the animals of a detected farm are culled and destructed and the premise is cleaned and disinfected.
- Tracing: Tracing activities involve the process of determining contacts that have left or entered a detected farm during a time interval preceding the detection (here, 2 months). The objective of tracing is to identify potential infectious contacts which may have introduced CSFV into the farm or spread CSFV to other farms.

4.3 Mathematical formulation of the model

In this Section, we describe in detail the Be-FAST model by presenting its general structure, the considered input parameters and the processes related to the CSFV spread and the control measures. The main notations used here are summarized in Table 4.2.

4.3.1 General description

The Be-FAST model is used to evaluate the spread of CSFV within and between farms into a specified region during a fixed time interval.

At the beginning of the simulation, the model parameters are set by the user¹. The parameters referring to farms and transports of pigs are detailed in Section 4.3.2. The

¹The value of the parameters used by Be-FAST should be set in function of the studied region (for example, due to the specific legislations). For instance, the parameters values considered during this work are adapted for their application to the province of Segovia (see Section 4.4). In particular the parameters referenced by [143] or [182] in Table 4.3 have been obtained by expert opinions of the Spanish administration. The reliability of those parameters is discussed in [186].

Table 4.2: Summary of the main notations used during this work to describe Be-FAST. Monte-Carlo is abbreviated by MC.

Notation	Description
M	Number of scenarios considered in the MC algorithm
SCE_m	m -th scenario of the MC algorithm
T	Maximum number of simulation days of each MC scenario
NF	Number of farms in the study region
$NP_i(t)$	Number of pigs in farm i at time t
$NSP_i(t)/NIP_i(t)$	Number of susceptible/infected pigs in farm i at time t
$SF/IF/TF/CF$	State of a farm: Susceptible/Infected/Infectious/Clinical Signs

other parameters are described in Sections 4.3.3-4.3.9 and are summarized in Table 4.3. Furthermore, the control measures presented in Section 4.2.3 are also implemented and can be activated or deactivated, when starting the model, in order to quantify their effectiveness to reduce the magnitude and duration of a CSF epidemic.

Be-FAST is based on a Monte-Carlo approach that generates $M \in \mathbb{N}$ epidemic scenarios (here, we call scenario to one iteration of the Monte-Carlo algorithm that simulates a possible evolution of the CSFV). More precisely, at the beginning (i.e., at time $t = 0$) of each scenario, denoted by SCE_m with $m = 1, 2, \dots, M$, all the farms are in the susceptible state except one farm, selected randomly by considering a discrete uniform distribution, which has one infectious pig and is classified as infectious. Then, during the time interval $[0, T]$, with $T \in \mathbb{N}$ the maximum number of simulation days, the within-farm and between-farm daily spread routines (described in Sections 4.3.3 and 4.3.4, respectively) are applied. Moreover, the daily processes simulating the detection by authorities of contaminated farms and the activated control measures (presented in Sections 4.3.8 and 4.3.9, respectively) are also run. If at the end of a simulation day all farms are in the susceptible state, the scenario SCE_m is stopped and we start the next scenario SCE_{m+1} . When the scenario SCE_M is over, many kinds of outputs can be generated. In Section 4.4.4, we present the most typical ones used to analyze the performance of an epidemiological model.

A diagram summarizing all those steps is presented in Figure 4.1.

Remark 2 *Currently, Be-FAST does not simulate the possible introduction or spread of CSFV due to contacts with foreigner regions. In fact, we simulate scenarios assuming that: the foreigner sources of CSF have been already identified and cannot send infected pigs to Segovia; and, once the index case in Segovia is detected, zoning is implemented and other regions do not buy animals from Segovia (thus, we neglect this way of infection). This is a limitation of Be-FAST which is common with other epidemiological models (such as the one described in [147]) and is discussed in [185].*

Table 4.3: Summary of the parameters used by the Be-FAST model²(except those referring to farms and transports of pigs which are presented in Section 4.3.2). From Left to Right: short parameter description; distribution (Dist.) or value considered during this work; and bibliographic reference.

Parameter description	Dist./Value	Reference
Transmission parameters of the SI model: $\beta_{\text{far}}/\beta_{\text{fat}}/\beta_{\text{ftf}}$	0.66/0.4/0.53	[156]
Daily PI due to local spread	See Table 4.4	[147]
PI due to infectious vehicles transporting pigs	Ber(0.011)	[251]
PI due to infectious INT vehicles	Ber(0.0068)	[251]
PI due to infectious SDA persons	Ber(0.0065)	[251]
Daily PD of the index case due to clinical signs	Ber(0.03)	[147]
Daily PD due to clinical signs	Ber(0.06)	[147]
PD due to tracing	Ber(0.95)	[182]
Maximum daily PD due to protection zone	Ber(0.98)	[143]
Maximum daily PD due to surveillance zone	Ber(0.95)	[143]
PR of animal movements of detected farms	Ber(0.99)	[143]
PR of INT movements of detected farms	Ber(0.95)	[143]
PR of SDA movements of detected farms	Ber(0.80)	[143]
PR of animal movements of zoned farms	Ber(0.95)	[143]
PR of INT movements of zoned farms	Ber(0.90)	[143]
PR of SDA movements of zoned farms	Bern(0.70)	[143]
PR of all movements of non zoned farms	Ber(0.40)	[143]
Probability of tracing animal movements	Ber(0.99)	[143]
Probability of tracing INT movements	Ber(0.70)	[143]
Probability of tracing SDA movements	Ber(0.40)	[143]
Duration of general movement restriction	30 days	[182]
Duration of protection/surveillance zones	30/40 days	[182]
Radius of protection/surveillance zones	3/10 (km)	[182]
Maximum number of traced farms per day	Poi(60)	[143]
Maximum number of depopulated farms per day	Poi(20)	[143]
Time to repopulate a depopulated farm	Poi(90) days	[143]
Time to depopulate a detected farm	See Table 4.5	[76]
Tracing period	60 days	[143]
Latent period	Poi(7) days	[148]
Incubation period	Poi(21) days	[148]
Daily number of contacts with INT vehicles of farms	Poi(0.4)	[148]
Number of farms visited by an INT vehicle per day	Poi(4)	[143]
Daily number of contacts with SDA persons of farms	Poi(0.3)	[148]
Number of farms visited by a SDA person per day	Poi(3)	[143]

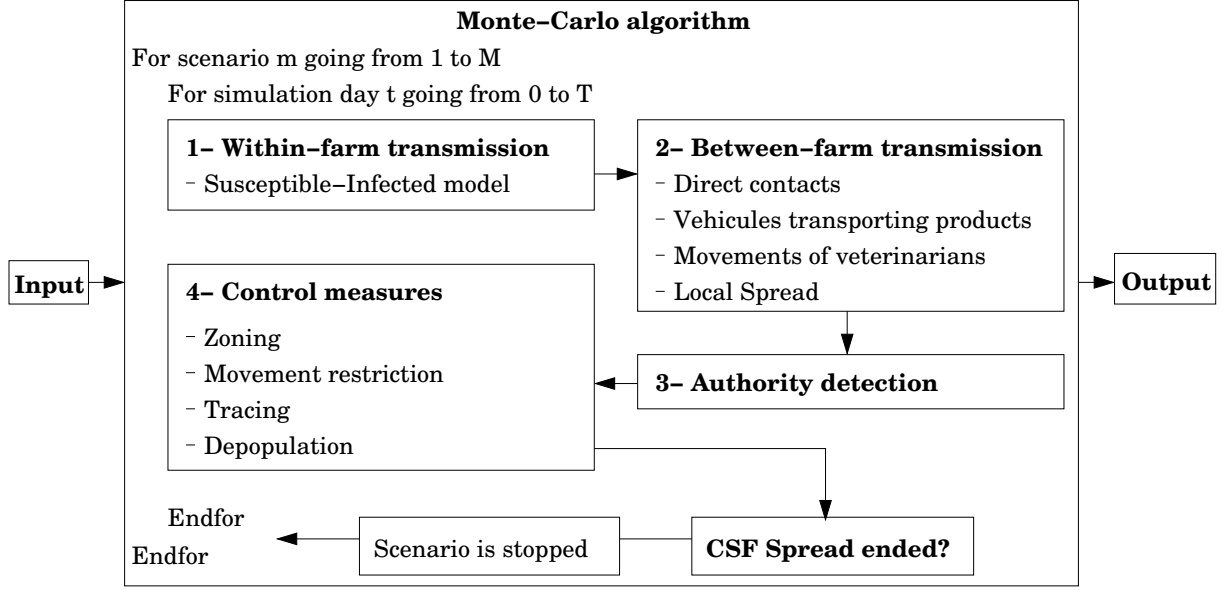


Figure 4.1: Diagram summarizing the Be-FAST model steps presented in Section 4.3.1.

4.3.2 Farms and transports of pigs inputs

We consider a study region containing $NF \in \mathbb{N}$ farms. For each farm, identified as farm number i (also called farm i) with $i = 1, \dots, NF$, the following data are given: the geographical coordinates of the farm centroid; $NP_i(0) \in \mathbb{N}$ the number of pigs at the first day of the simulation (i.e., $t = 0$); the type of pig production of the farm (here, farrowing, fattening or farrow-to-finish) [156]; the Integrator (INT) group (i.e., group of farms who share material and vehicles transporting products) identifier; the Sanitary Defense Association (SDA) group (i.e., group of farms who share veterinarians) identifier.

Furthermore, the following information of all farm to farm pig shipments occurring during a specified time interval is also provided: the number of pigs shipped; the date of the shipment; and the farms of origin and destination of the shipment.

4.3.3 Within-farm CSFV spread

The daily CSFV spread within a particular contaminated farm i is modeled by using a discrete time stochastic Susceptible-Infected (SI) model [32, 156]. We assume that the pigs in this farm are characterized to be in one of those two states: Susceptible or Infected, described in Section 4.2.1. In order to reduce the computational complexity of our model (see Remark 3), the Infectious and Clinical Signs states are simulated only at the farm level (more details are given in Section 4.3.4). The natural pig mortality is also neglected [185].

Under those assumptions, the evolution of $NSP_i(t)$ and $NIP_i(t)$, denoting the number

²In Table 4.3, we consider the following abbreviations: PI=Probability of Infection; PD=Probability of Detection; PR=Probability of Restriction; SDA=Sanitary Defense Association group; INT=Integrator group; Poi(X)=Poisson distribution with mean X; Ber(X)=Bernoulli distribution with mean X.

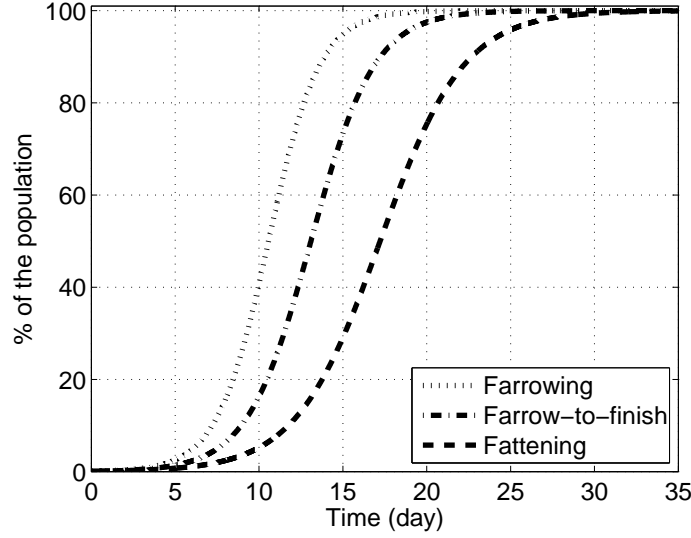


Figure 4.2: Time evolution of the percentage of infected pigs obtained by considering System (4.3.1) and a farm containing 1000 pigs and with one infected pig at time $t = 0$, in function of the farm type: farrowing, fattening and farrow-to-finish.

of susceptible and infected pigs in farm i at time t , respectively, is given (in a continuous version) by

$$\frac{dNSP_i(t)}{dt} = -\beta_i \frac{NSP_i(t)NIP_i(t)}{NP_i(t)}, \quad \frac{dNIP_i(t)}{dt} = \beta_i \frac{NSP_i(t)NIP_i(t)}{NP_i(t)}, \quad (4.3.1)$$

where $NP_i(t) = NSP_i(t) + NIP_i(t)$ is the number of pigs in farm i at time t , $\beta_i \in \mathbb{R}$ is the transmission parameter set to $\beta_{\text{far}} = 0.66$, $\beta_{\text{fat}} = 0.40$ or $\beta_{\text{ftf}} = 0.53$ depending on the farm type: farrowing, fattening or farrow-to-finish pig farms, respectively [156]. The evolution of the proportion of infected pigs governed by System (4.3.1) and obtained by considering a farm containing 1000 pigs and with one infected pig at time $t = 0$, in function of the farm type, is presented in Figure 4.2.

Moreover, we are interested in obtaining integer values of infected and susceptible pigs in farms and in introducing some randomness in System (4.3.1) (the within-farm CSFV spread may be slightly different from one farm to another) but respecting its general behavior. Thus, we consider the following discrete version of System (4.3.1) [156]

$$NSP_i(t+1) = NSP_i(t) - \min(\Gamma_i(t), NSP_i(t)), \quad NIP_i(t+1) = NIP_i(t) + \min(\Gamma_i(t), NSP_i(t)),$$

where t corresponds to the simulation day and $\Gamma_i(t) \in \mathbb{N}$ follows a Poisson distribution with mean $\beta_i(NSP_i(t)NIP_i(t))/NP_i(t)$.

Remark 3 *Although the SI model presented in this Section seems to be too simple (with only two pig states) to simulate the within-farm CSFV spread, in practice it gives a good ratio between spread modeling accuracy and computational time. Indeed, we have to consider that this model is applied to each farm infected during a Monte-Carlo scenario, which*

can dramatically increase the computational time needed by the Be-FAST model. For example, we have tried to consider the Infectious and Clinical Signs states at the pig level. In that case, we have obtained results similar to the ones given by the model described here (2% of variation) with a significant increase of the computational time (+30%). For the same reasons, considering a model more complex than a SI model (such as one that simulates the spatial diffusion of the CSFV within a farm) does not appear to be a reasonable choice in terms of model efficiency (for instance, if an epidemiological model is used to support decisions in cases of real outbreaks, its outputs should be given within a day [143]).

4.3.4 Between-farm CSFV spread

The CSFV spread between farms is modeled by using a spatial stochastic Individual-Based model [61]. In this model, farms are classified in one of those four states (see Section 4.2.1): Susceptible (SF), Infected (IF), Infectious (TF) and Clinical signs (CF). The daily transition from a particular farm state to another one is modeled by considering the direct and indirect contacts and the natural evolution of the CSF presented in Sections 4.2.1 and 4.2.2. Those transition processes are described in Sections 4.3.5-4.3.7.

4.3.5 State transition due to direct contacts

The CSFV spread by direct contacts is assumed to occur due to the movements of infected pigs between farms. Those movements are estimated by using the data of the shipments of pigs introduced in Section 4.3.2. Although this database usually contains previous years' information (see Section 4.4.2), the transports of pigs are similar from one year to another with slight variations [143]. Thus, we have decided to use random movements, generated from the pig transports data, instead of using the exact ones (using exact or random shipments does not impact dramatically the model outputs, see [186]). Due to the construction process described below, those random movements exhibit similar characteristics (i.e., farms of origin and destination, date and number of moved pigs) than the real ones. However, this process allows to consider, with a low probability, transports to or from farms not included in the shipment database (for instance, due to errors). More precisely, at each simulation day t , those shipments are simulated as following:

We compute the number of movements occurring during the simulation day t by considering a Poisson distribution with mean being the number of all movements occurring at day t in our database. Then, for each simulated movement:

- We select randomly the farm of origin of the movement $i \in [1, \dots, NF]$ and the farm of destination of the movement $j \in [1, \dots, NF]$, with $i \neq j$, by considering the discrete probability $\mathbb{P}_M((i, j) = (k, l)) = M_{\text{mov}}(k, l) / (\sum_{m=1}^{NF} \sum_{n=1, n \neq m}^{NF} M_{\text{mov}}(m, n))$, where k and $l \in [1, \dots, NF]$, $k \neq l$ and $M_{\text{mov}}(k, l) \in \mathbb{R}$ is the number of movements from farm k to farm l in the database plus 10^{-6} (to take into account with a small probability possible movements not occurring in our database). We note that \mathbb{P}_M is computed once before the simulations and only each time we get a new database (other parameters related to the database may be also calculated once before running the model).

- We compute $w_{(i,j)} = \min \{ \text{Ceil}(\overline{w_{(i,j)}}(NP_i(t)/NP_i(0))), NP_i(t) \}$, the number of pigs shipped during this movement from farm i to farm j . In the previous expression, $\overline{w_{(i,j)}} \in \mathbb{R}$ denotes the mean number of pigs moved when considering all the shipments between farms i and j in our database and $\text{Ceil}(x)$ returns the nearest integer greater than or equal to $x \in \mathbb{R}$. In the case of no movement from farm i to farm j in the database, $\overline{w_{(i,j)}}$ is set to the mean number of pigs moved taking into account all the database shipments.
- Next, we move $w_{(i,j)}$ pigs from the origin farm i to the destination farm j . Those pigs are selected randomly in $NSP_i(t)$ and $NIP_i(t)$, considering a discrete uniform distribution. We denote by $w_{(i,j)}^S \in \mathbb{N}$ and $w_{(i,j)}^I \in \mathbb{N}$ the number of susceptible and infected pigs that are moved during this simulated shipment, respectively. Thus, the evolution of pigs in farm i and j is governed by $NSP_i(t) = NSP_i(t) - w_{(i,j)}^S$, $NIP_i(t) = NIP_i(t) - w_{(i,j)}^I$, $NSP_j(t) = NSP_j(t) + w_{(i,j)}^S$ and $NIP_j(t) = NIP_j(t) + w_{(i,j)}^I$.
- Finally, if $w_{(i,j)}^I > 0$, the state of farm j is set to the state of farm i in one of the following cases: The state of farm j is SF ; the state of farm j is IF and the state of farm i is TF or CF ; the state of farm j is TF and the state of farm i is CF . In other cases, the state of farm j remains unchanged.

4.3.6 State transition due to indirect contacts

As specified in Section 4.2.2, the CSFV spread due to indirect contacts is assumed to occur by either movements of vehicles transporting pigs, movements of INT vehicles (i.e., vehicles transporting products from or to farms in the same INT group), movements of SDA persons (i.e., persons visiting farms in the same SDA group) or the so called local spread (i.e., spread due to contacts with the neighborhood which include: airborne spread and contaminated persons or fomites in the vicinity).

In Paragraphs *A-C*, we describe those four kinds of indirect contacts and the way they contribute to the CSFV spread from farm to farm. Then, in Paragraph *D*, we show how this spread affects the farms at the levels of pig numbers and state.

A- Movements of vehicles transporting pigs: We consider the pig shipments generated in Section 4.3.5. If the farm of origin of the transport is either in the state TF or CF , the truck transporting pigs is considered as contaminated and, thus, can infect the farm of destination. In that case, we assume that the probability of CSFV infection in the farm of destination due to the contact with the contaminated vehicle is modeled by using a Bernoulli distribution with mean 0.011 [251].

B- Movements of INT vehicles and SDA persons: The CSFV spread by contact with INT vehicles (SDA persons, respectively) visiting farms is assumed to occur only among the farms belonging to the same INT group (SDA group, respectively) and with the following assumptions:

- The daily number of contacts with INT vehicles (SDA persons, respectively) per farm is assumed to be Poisson distributed with a mean of 0.4 (0.3, respectively) [148].
- An INT vehicle (SDA person, respectively) can visit a maximum of 4 (3, respectively) farms per day [143].
- An INT vehicle (SDA person, respectively) is contaminated if, previously, it has visited a spreading farm (i.e., a farm either in the state TF or CF , see Section 4.2.1) [147, 251].
- The probability of CSFV infection in a farm per contact with a contaminated INT vehicle (contaminated SDA person, respectively) is modeled by using a Bernoulli distribution with mean 0.0068 (0.0065, respectively) [251].

Thus, for each simulation day and for each integrator group INT, we build the routes of the INT vehicles and we simulate the way they spread CSFV by considering the process described below:

- For each farm in INT, we compute the number of INT vehicles visiting it by using a Poisson distribution with mean 0.4.
- Then, we list the farms that are visited by the INT vehicles and we rearrange this list, denoted by L_{INT} , randomly (taking into account that a farm cannot be visited twice consecutively).
- Next, a first INT vehicle visits the first four farms in L_{INT} , following the list order. Each fourth farm, until the end of L_{INT} , we consider a new INT vehicle (non-contaminated) starting from the next farm in L_{INT} .
- During each simulated trip, an INT vehicle becomes contaminated at the moment it visits a spreading farm and can infect other farms by considering a Bernoulli distribution with mean 0.0068.

The same method is used to model the itineraries and the CSFV spread of SDA persons, but with the corresponding parameters values.

C- Local spread: We assume that the CSFV local spread occurs to farms in the proximity of a farm either in the state TF or CF . It is mainly due to the airborne spread and contacts with contaminated neighborhood persons or fomites.

During this work, the daily probability of CSFV infection in farm j , due to the local spread from spreading farm i at simulation day t , is modeled by considering a Bernoulli distribution with mean $\overline{PIL}(d(i, j)) \times (NIP_i(t)/N(0))$, where $N(0) = (\sum_{i=1}^{NF} NP_i(0))/NF$ is the mean number of pigs per farm at day 0, $d(i, j)$ is the distance (in meters) between the centroid of farms i and j and $\overline{PIL}(x) \in [0, 1]$ is the mean daily probability of CSFV infection due to local spread between two farms at a distance x . Moreover, $\overline{PIL}(x)$ is built by interpolating the data presented in Table 4.4 [147].

Table 4.4: Interpolation points used to compute the function \overline{PIL} [147].

Distance x (meters)	0	150	250	500	1000	2000
Value of $\overline{PIL}(x)$	0.02	0.014	0.009	0.0038	0.0019	0

D- New infection and state transition: For each new CSFV infection occurring at day t in farm i during the processes described in Paragraphs *A* to *C*, if $NSP_i(t) \geq 1$, we infect one new pig in farm i by considering $NSP_i(t) = NSP_i(t) - 1$ and $NIP_i(t) = NIP_i(t) + 1$. Furthermore, if the state of farm i is SF , we change it to IF .

4.3.7 State transition due to the CSF natural evolution

According to the characteristics of the CSF evolution described in Section 4.2.1, we consider the following changes in the farm state: when a farm reaches the state IF (or TF), it will pass at state TF (CF) after a latent period (incubation period) that follows a Poisson distribution with mean 7 (21) days [148].

4.3.8 Detection of contaminated farms

As specified in Section 4.2.3, a contaminated farm is generally detected by the observation of clinical signs of its pigs (i.e., the farm is in state CF) [158]. Thus, for each farm in the state CF , its daily probability of detection is modeled by using a Bernoulli distribution with mean 0.03 (0.06, respectively), before (after, respectively) the detection of the index case [140, 147]. Furthermore, a contaminated farm can be also detected due to the control measures presented in Section 4.3.9.

4.3.9 Control measures

We now describe the control measures, introduced in Section 4.2.3, implemented in Be-FAST.

4.3.10 Movement restrictions

A drastic restriction on movements (outgoing or incoming in farms) is applied to detected farms. Reductions on the transports of animals, the INT vehicles movements and the SDA persons movements in the detected farms are assumed to be Bernoulli distributed with a mean of 0.99, 0.95 and 0.8, respectively.

Furthermore, after each detection, a general movement restriction considering those three kinds of movements, which follows a Bernoulli distribution with mean 0.4, is applied to all farms³ during 90 days [143, 182].

³This control measure is adapted for studying CSFV spread in the province of Segovia (see Section 4.4). For larger areas (e.g., a country), the movement restrictions should be limited to a part of the studied region.

Table 4.5: Probability of the number of days to wait before depopulating a detected farm [76].

Number of days	0	1	2	3	4	5	6	7	8
Probability	0.11	0.58	0.2	0.06	0.04	0.004	0.003	0.0015	0.0015

4.3.11 Zoning

The farms at a distance of less than 3 km from a detected farm are set in a protection zone during 30 days, whereas the farms at a distance between 3 km and 10 km from a detected farm are set in a surveillance zone during 40 days [182]. Overlapping of the period of a farm in protection and surveillance zones is allowed (i.e., if a farm is already within a zone, we add the days of the new zone to those of the old zone).

A movement restriction is applied to farms within those zones (called zoned farms) [182]. Pig transports, movements of INT vehicles and movements of SDA persons are randomly reduced by considering a Bernoulli distribution with mean 0.95, 0.9 and 0.7, respectively [182, 143]. Furthermore, we apply another surveillance process to zoned farms, in addition to the one described in Section 4.3.8. The daily probability of detection of farm i in the state CF due to this new surveillance is assumed to be dependent on the proportion of infected animals and is modeled by considering a Bernoulli distribution with mean $(\alpha \times NIP_i(t))/NP_i(t)$, where $\alpha = 0.98$ ($\alpha = 0.95$, respectively) if farm i is within a protection (surveillance, respectively) zone [143].

4.3.12 Depopulation

The depopulation (i.e., the culling of all animals) of detected farm i occurs after a random time period, generated by using the data provided by Table 4.5 [76], starting from the day of its detection. However, the maximum number of farms that can be depopulated per day is assumed to follow a Poisson distribution with mean 20 [143]. Thus, if this limit is reached, the farm is depopulated the following days. When farm i is depopulated, its number of pigs is set to 0 and it is not considered anymore by the model. Then, after a period following a Poisson distribution with mean 90 days [182], the farm is repopulated (i.e., new pigs are introduced): its number of susceptible pigs is $NP_i(0)$, its state is set to SF and it is again taken into account by Be-FAST.

4.3.13 Tracing

The objective of tracing is to identify infectious contacts which may have introduced CSFV into a detected farm or spread CSFV to other farms. We include the tracing of all contacts (i.e., farms sending/receiving animals or sharing SDA persons/INT vehicles) of a detected farm occurring 60 days before the detection [182]. However, due to registration failures (e.g., errors in databases), tracing all the contacts is not always possible.

More precisely, when farm i is detected, we list all the farms who have shared, 60 days before the detection, at least one INT vehicle, one SDA person or one animal shipment with this farm i . Then, for each farm in this list, we decide if it is traced or not according

to the probabilities of tracing a farm due to animal transport, INT vehicle movement or SDA person movement which are assumed to be Bernoulli distributed with a mean of 0.99, 0.7 and 0.4, respectively [143]. Next, for each farm to be traced, we select its day of tracing taking into account, as in Section 4.3.12, that the maximum number of traced farms per day is governed by a Poisson distribution with mean 60. Finally, we perform a detection process to the traced farms, the day of their tracing, by considering that the probability of detecting a contaminated traced farm follows a Bernoulli distribution with mean 0.95 [182].

4.4 Model Validation

In order to validate our model, we perform various numerical experiments, described in Section 4.4.1. The results obtained by Be-FAST, presented in Section 4.4.7, are compared with those generated by two other epidemiological models considering similar experiments (see Section 4.4.5) and with data observed during real CSF outbreaks (see Section 4.4.6).

4.4.1 Numerical experiments

In the following, we present the numerical experiments used for the model validation. In particular, we detail the inputs, the scenarios parameters and the considered outputs.

4.4.2 Farms and pig transports inputs

We consider the Spanish province of Segovia⁴, one of the most important pig production areas in Spain, which has a surface of 6796 km². We use real databases of the year 2008, provided by the Spanish Regional Government of Castilla and Leon [143] and the Spanish Ministry of the Environment and Rural and Marine Affairs [182], corresponding to the inputs described in Section 4.3.2. In 2008, $NF = 1400$ pig farms, containing a total of 1403800 pigs, were located in Segovia. Those farms were divided in 11 INT groups and 34 SDA groups. Furthermore, 208 farms were of the type farrowing, 510 fattening and 682 farrowing-to-finish. Finally, during this year, there were 10046 pig shipments. The locations of those pig farms and the province of Segovia are shown in Figure 4.3.

4.4.3 Scenarios parameters

We have considered two kinds of simulations:

In the first one, we do not consider any control measure and we run the model during $T = 200$ days (this value gives a good ratio between results precision and computational time, see [185, 186]). This case is denoted by **NM** (No Measure). The interest of this experiment is to evaluate the principal routes of CSFV spread.

⁴This reduced region has been already used to illustrate, with success, the behavior of our model in [185, 186] and, thus, is also applied here to validate our model. However, this region could be replaced by any larger region when the required databases are available.

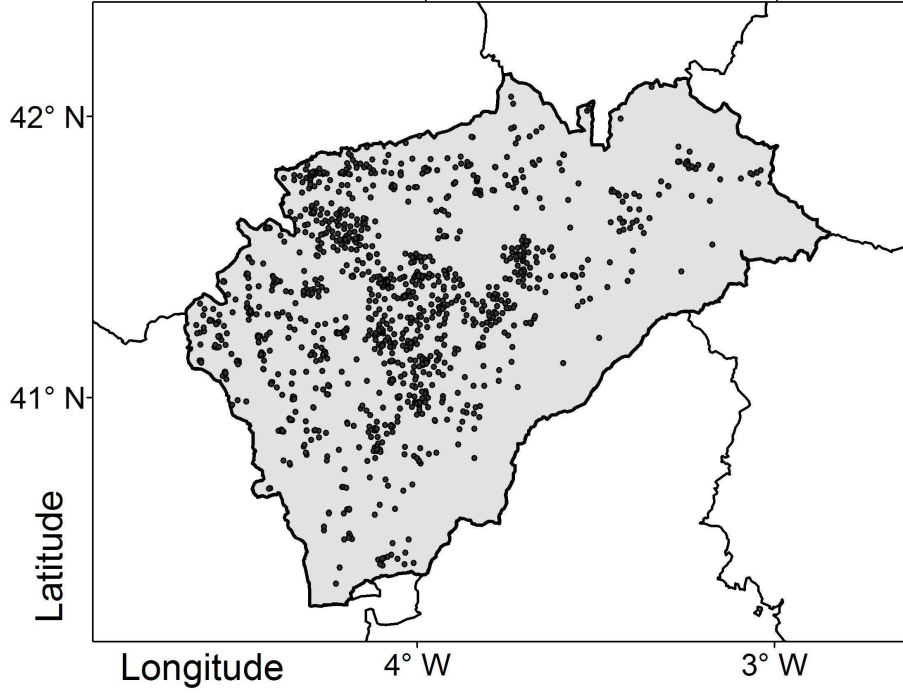


Figure 4.3: Coordinates and boundaries of the province of Segovia (in gray). The locations of the considered pig farms are represented by black spots (\bullet).

In the second one, all the control measures described in Section 4.3.9 are activated and the model is run during a maximum period of three years (i.e., $T = 1095$ days), which is large enough to ensure the end of a CSF epidemic [185]. This case is denoted by **WM** (With Measures). In this experiment, which is more realistic and classical (i.e., this experiment is also considered in other works, such as [149] and [180]) than **NM**, we are interested in evaluating the magnitude of the epidemic and the efficiency of the control measures.

In both cases, we set $M = 1000$ scenarios. This value gives a good compromise, in the particular experiments considered here for the Be-FAST model, between the stability (with variations less than 3%) of the outputs presented in Section 4.4.4 (including risk maps) and the computational complexity [186]. Furthermore, we want to point out that, in the literature, some models are run with a much lower value. For instance, in [140, 147, 148, 149], the number of scenarios of the considered Monte-Carlo algorithm is 100. This reinforces the idea of precision of the results obtained during this work. We also remark that this value of M is adapted to our particular number of farms $NF = 1400$. When considering a greater value of NF , M should be increased proportionally [177].

4.4.4 Considered outputs

When the scenario SCE_M is over, many types of outputs can be obtained. Here, we consider the most typical ones used to evaluate the performances of an epidemiological model [140, 149, 187].

More precisely, for each scenario SCE_i with $i = 1, \dots, M$, we compute: the number of infected farms, denoted by NIF ; and the duration of the epidemic (i.e., the number of days between the beginning and the ending of the scenario), denoted by DUR . For those both quantities, we calculate, regarding all the scenarios, their mean, minimum and maximum values; their 95% prediction interval; their quartiles; and their discrete distribution functions.

In addition, taking into account the M scenarios, we evaluate: the percentages of infection due to local spread, INT vehicles, SDA persons and transports of pigs; and the percentages of detection of contaminated farms, after detecting the index case, due to observation of the clinical signs, zoning and tracing.

Furthermore, for each farm i , we compute its risk of CSFV introduction, denoted by $RI(i)$. It is defined as the number of times that farm i becomes contaminated during the whole Monte-Carlo simulation. In particular, in order to identify the risk zones in the studied region, we are interested in obtaining the geographical distribution of RI . Typically [197], the risk zones are classified in three categories: high, medium and low risk. This is useful, for instance, to design preventive control measures to fight CSFV [83]. To do so and to compare the values of RI given by the models presented in Section 4.4.5, we first normalize $RI(i)$ by considering $\bar{RI}(i) = \hat{RI}(i) / (\max_i \hat{RI}(i))$, where $\hat{RI}(i) = RI(i) / (\sum_i RI(i))$. Then, we obtain the spatial distribution of \bar{RI} in Segovia by interpolating the values of $\bar{RI}(i)$ considering an inverse distance weighted method. Finally, the identification of the three risk zones is done by applying the Jenks Natural Breaks (JNB) classification method. Those both last steps are done by using ArcGIS 9.1 [79].

4.4.5 Comparison with other models

In order to validate the Be-FAST model, we perform the experiments presented in Section 4.4.1 by using the two following models:

A) Be-FAST model (BF): A Matlab 7.8 [190] implementation of Be-FAST.

B) InterSpread Plus model (IS): We also consider InterSpread Plus 1.0.49.5 [188], a commercial C++ implementation of a state transition Individual-Based model presented in [244]. It is one of the most popular epidemiological modeling software used worldwide. However, in our opinion, it has several drawbacks, such as the scarce transparency of the code (i.e., it is a black-box program) and the difficulty to incorporate complex databases with real farm-to-farm movements or contacts.

We intend to reproduce the same processes as the one used by **BF**. As it is out of the scope in this Chapter, we do not describe the **IS** model in detail, but we only present here the differences between both models (other parts are similar).

IS does not allow simulating the within-farm transmission (it is a purely between-farm spread model), so it is not possible to compute the number of infected or susceptible pigs of a particular farm i at a given time. For this reason, the model coefficients cannot be expressed as a function of $NSP_i(t)$ or $NIP_i(t)$. Thus, all the coefficients depending on those numbers are set to constant values: the daily probability of infection due to local

Table 4.6: Weight coefficients of the contaminated farms, in function of the number of days after the farm infection, used in our simulations with the InterSpread Plus model.

Number of days	0	5	10	13	15	20	22
Weight coefficient	0.001	0.03	0.17	0.55	0.7	0.95	1

Table 4.7: Probability of selecting a farm of destination for pig shipments, in function of the distance (in kilometers) with the farm of origin, used in our simulations with the InterSpread Plus model.

Distance interval (kilometers)	[0-15]	[15-30]	[30-45]	[45-60]	[60-80]	> 80
Probability of selection	0.4	0.29	0.18	0.07	0.04	0.02

spread is set to $\overline{PIL}(x)$ without interpolation; the probability of infection due to a pig shipment coming from a contaminated farm is 1; the daily probabilities of detection of a contaminated farm in a protection or surveillance zone follow a Bernoulli distribution with mean 0.98 and 0.95, respectively. However, for each contaminated farm i , **IS** allows to assign to this farm a weight coefficient, depending on the number of days after the farm infection, that is multiplied to all probabilities of infection due to contact with farm i . This process simulates the fact that the infectiousness of a farm increases with the time. The weight coefficients are reported in Table 4.6 and fit the SI evolution of a farrow-to-finish farm, depicted in Figure 4.2.

Moreover, the real commercial network (i.e., pig shipments, SDA and INT groups) cannot be integrated directly in **IS**. First, the swine transport process is simulated as follows. For each simulation day and farm i , we have to compute the number of pig transports sent by this farm. This is done by considering a Poisson distribution with mean given by the daily mean number of pig transports sent by farm i . Then, for each shipment, we select randomly a farm of destination according to the distance with farm i and the probability distribution given in Table 4.7. The values of the daily mean number of pig transports sent by each farm and Table 4.7 are obtained from our database. Secondly, the SDA and INT contacts are simulated as in our model but considering only one SDA and INT group. The farms to be visited are randomly selected in function of their distances according to Table 4.7. From those simplifications, we see that **IS** does not allow to incorporate the real commercial contacts between farms.

Finally, as a last difference with **BF**, **IS** considers that when a farm is infected by a pig shipment its state is set to IF .

Furthermore, we also compare some results obtained by **BF** and **IS** with those given by the model described below on a region different than Segovia by considering experiments similar to the ones presented in Section 4.4.1.

C) Kartsen et al. model (KM): This Individual-Based model, presented in [147, 148], simulates only the between-farm spread and is applied to a fictitious German region,

but based on real statistics, in [149]. The considered region, which has an area of 2230 km², contains 2986 farms (1896 fattening farms, 543 farrowing farms, 546 farrow-to-finish farms). No data on the commercial network between farms are considered. The simulated processes for the CSFV spread and the control measures are similar to the ones used by **BF** (except the spread due to artificial insemination which is neglected in **BF** and **IS** due to its low probability of infection, see [252]). Furthermore, the model parameters are adapted to the German country and can be found in [147]. However, they are close to the ones used in this work. The interesting results obtained by **KM**, reported in [149], are the mean, minimum and maximum values of the variables *NIF* and *DUR* for the **WM** experiment.

4.4.6 Comparison with real epidemic data

In order to compare the validity of the results given by **BF** and **IS**, we consider data observed during the following two real CSF epidemics occurring in 1997-98:

A) Segovia: A description of this epidemic can be found in [184] and data were provided by [143]. During this event, 22 farms were infected and the epidemic duration was approximatively 60 days. As Segovia is the region studied during our experiments, we use the spatial location of those infected farms to validate the risk maps generated by **BF** and **IS**. However, data regarding the proportions of infection routes and control measures efficiency are not available. Thus, we have to consider another real case.

B) The Netherlands: This case, detailed in [76, 140], is interesting as this country has similar pigs production characteristics than Segovia. Due to the high epidemic magnitude (429 infected farms) and the quality of the experiments done in situ, the results provided in this literature are assumed to be valid reference values. Here, we are interested in the proportions of infection due to each CSFV route and detection of each control measure observed during this epidemic and reported in Table 4.8 in the **Real** row.

Remark 4 *We note that real epidemic data should be considered as only one possible evolution of the disease. However, they give some interesting reference values in order to interpret the output given by simulation models. Of course, this interpretation should be considered as an indicator of the validity of a model but not as a rigorous fact. This approach is assumed to be appropriate in literature (for instance, see [140, 149]).*

4.4.7 Results

All the simulations presented in this Section are run on a computer with a 2.4Ghz Core2 Duo P8600 CPU, 4Gb of DDR3 memory and Windows Vista 32bit OS. The outputs, described in Section 4.4.4, obtained during those experiments are reported in Tables 4.8-4.9 and, some of them, are depicted in Figures 4.4-4.5.

The computational time needed by models **BF** and **IS** to solve the **NM** and **WM** experiments are presented in Table 4.8-(**Comp. Time** columns). We can see that the **IS** model is the fastest one. In particular, for the **NM** case (with highest numbers of infected

Table 4.8: Results obtained when solving the **NM** (No Measure) and **WM** (With Measures) experiments considering models **BF** (Be-FAST) and **IS** (InterSpread Plus): computational time (**Comp. Time**), in seconds, needed to solve each case; percentages of infection due to each CSFV route (**Route**) obtained by considering the **NM** case: local spread (*LS*), Integrator vehicles (*INT*), Sanitary Defense Association persons (*SDA*) and transport of animals (*TA*); percentages of detection of contaminated farms due to each control measure (**Measure**) obtained by considering the **WM** case: observation of clinical signs (*CS*), zoning (*ZO*) and tracing (*TR*). Row **REAL** corresponds to the percentages observed during the real epidemics occurring in 1997-98 in The Netherlands [76, 140].

Model	Comp. Time		Route				Measure		
	NM	WM	<i>LS</i>	<i>INT</i>	<i>SDA</i>	<i>TA</i>	<i>CS</i>	<i>ZO</i>	<i>TR</i>
BF	28000	14500	54	26	14	6	47	30	23
IS	4000	11000	51	13	10	26	38	50	12
REAL	-	-	52	24	20	4	55	27	18

Table 4.9: Some statistical values on the number of infected farms (*NIF*) and the epidemic duration (*DUR*) obtained by considering the **NM** (No Measure) and **WM** (With Measures) experiments and models **BF** (Be-FAST) and **IS** (InterSpread Plus): mean value (Mean); minimum value (Min.); maximum value (Max.); 95% Prediction Interval lower (PI[2.5%]) and upper (PI[97.5%]) bounds; and quartiles (Q1, Q2 and Q3). Some data available for the model **KM** in the **WM** case are also reported [149].

Model	Output	Mean	Min.	PI[2.5%]	Q1	Q2	Q3	PI[97.5%]	Max.
NM									
BF	<i>NIF</i>	32	1	1	4	16	40	122	339
IS	<i>NIF</i>	58	1	1	7	33	84	255	523
WM									
BF	<i>NIF</i>	3.3	1	1	1	2	3	14	53
BF	<i>DUR</i>	63	14	25	38	51	80	178	428
IS	<i>NIF</i>	4.6	1	1	1	1	3	34	68
IS	<i>DUR</i>	79	14	28	38	54	81	326	729
KM	<i>NIF</i>	7.5	1	-	-	-	-	-	56
KM	<i>DUR</i>	84	20	-	-	-	-	-	230

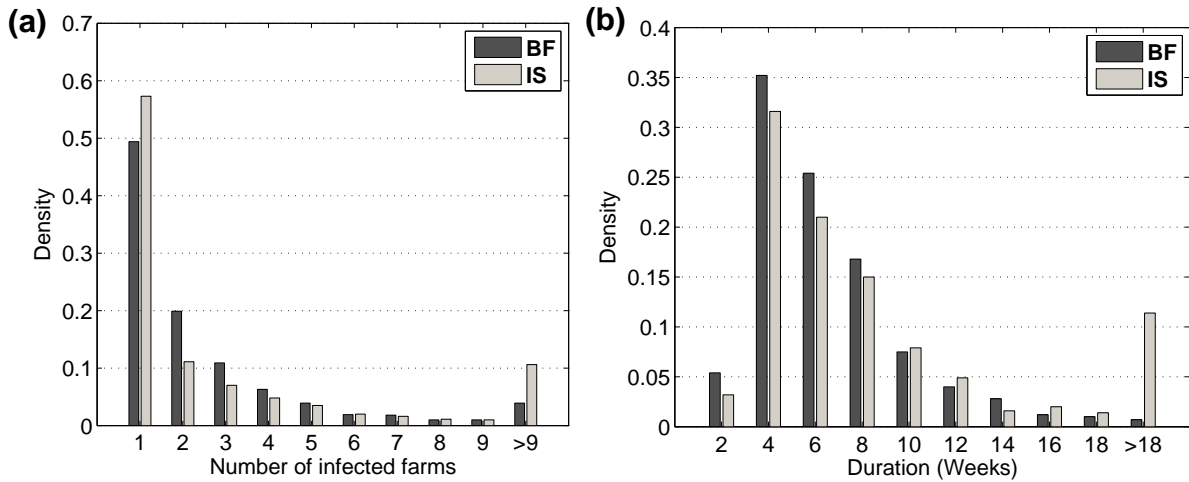


Figure 4.4: Discrete distributions of (a) the number of infected farms (*NIF*) and (b) the epidemic duration (*DUR*) (in weeks) obtained by models **BF** (Be-FAST, dark gray) and **IS** (InterSpread Plus, light gray) for the **WM** (With Measures) experiment.

farms), the difference between both models is quite high (**BF** is 7 times slower). In the **WM** case, which is more realistic, the difference is reasonable. This can be explained, in part, by the fact that our model has a more complex and complete structure than **IS** (for instance, the use of SI model for each contaminated farm, dynamic coefficients, etc.), and therefore, requires more computations. Another explanation is the choice of the program languages used by **BF** and **IS**. Our model is implemented in Matlab script, an interpreted language known to be slow in comparison to compiled languages [65], such as C++, which is used to implement **IS**. Matlab was chosen in order to obtain quickly a first implementation of our model and for the easiness to process the outputs. The computational time required by **BF** could be significantly decreased by programming it in C++.

The percentages of infection in function of the CSFV routes are given in Table 4.8- (**Route** columns). We can see that both models identify the local spread as the main source of infection. Moreover, the proportions due to SDA persons are quite similar in the two cases. The main differences between models **BF** and **IS** are obtained when regarding the transports of animals and the INT vehicles proportions. In that case, **IS** considers the animal shipments as the second most important cause of CSFV infection instead of the INT vehicles. This discordance may be due to the fact that **IS** does not take into account the number of infected pigs when performing the animal transports, whereas **BF** uses this information. Thus, when the number of infected animals in the farm of origin of the transport is low, many of the pig shipments simulated by **BF** do not infect the destination farms, decreasing the proportion of infection due to this route. When we compare those results with the proportions observed during the 1997-98 epidemic in The Netherlands [140], we see that the **BF** outputs fit better those real data (in particular the *INT* and *TA* columns) than the **IS** ones.

The percentages of detection of contaminated farms in function of the control measures

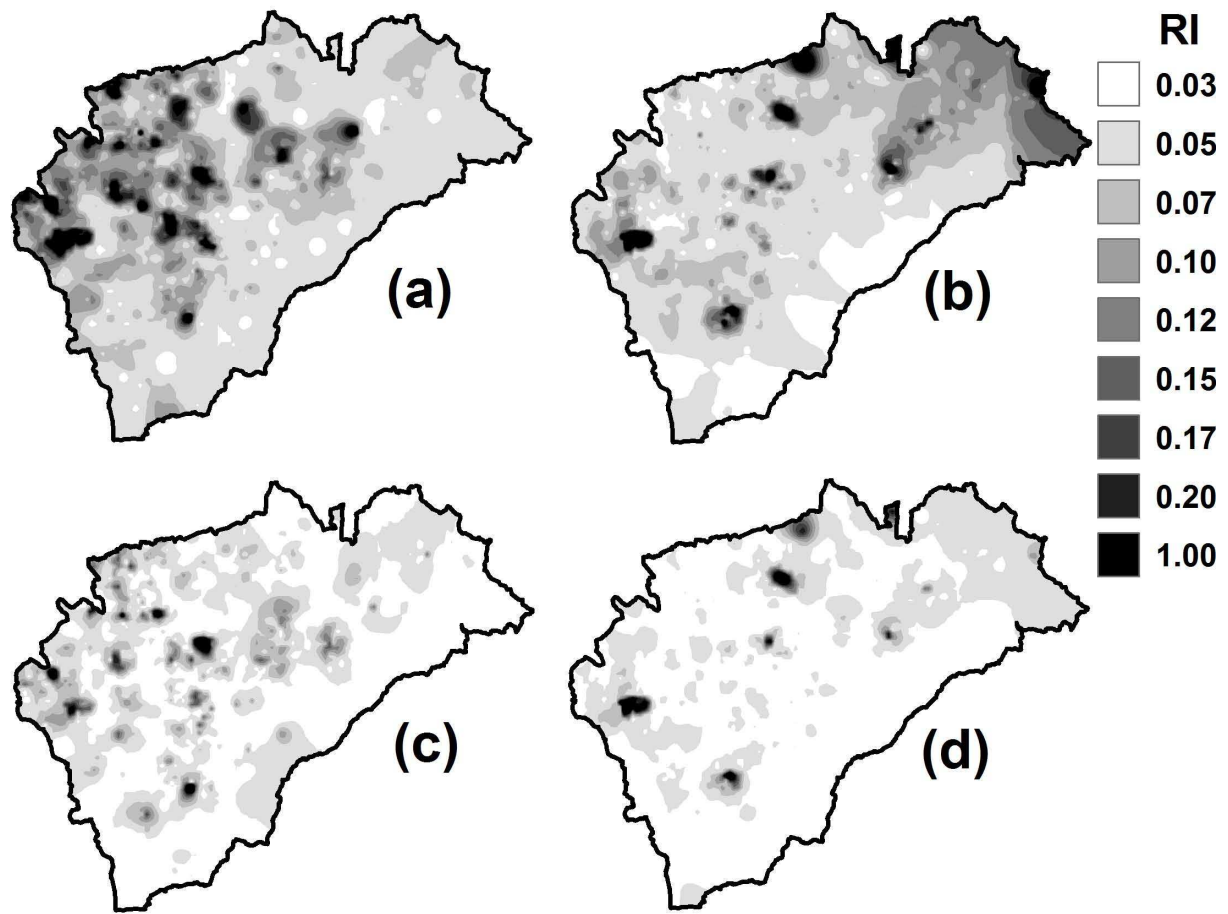


Figure 4.5: Normalized risk of CSFV introduction (\bar{RI}) interpolated maps obtained by models (a) **BF** (Be-FAST) for the **NM** (No Measure) case, (b) **IS** (InterSpread Plus) for the **NM** case, (c) **BF** for the **WM** (With Measures) case and (d) **IS** for the **WM** case. The grayscale colormap corresponding to the considered Jenks Natural Breaks classification, described in Section 4.4.4, is also reported.

are reported in Table 4.8-(**Measure** columns). On one hand, **IS** considers that the zoning is the most efficient control measure, then the observation of the clinical signs and, finally, the tracing. On the other hand, **BF** returns the observation of clinical signs as the best detection technique, the zoning and the tracing presenting similar efficiency. This can be explained by the fact that, as described in Section 4.3.11, **BF** uses the number of infected animals in zoned farms to generate the probabilities of detection due to the zoning process (thus, the efficiency of this control measure may be reduced for farms with a low proportion of infected animals), whereas **IS** does not allow this possibility. As previously, the percentages generated with **BF** are closer to the real data reported during the 1997-98 epidemic in The Netherlands [76] than those given by **IS**.

Both results in the proportions of infection and detection seem to indicate that our approach, which consists in simulating the number of infected animals in farms and using it in the formulas of the **BF** model coefficients, provides better results than not considering it and is suitable for generating epidemiological models presenting a realistic behavior.

The statistical values associated to the number of infected farms (*NIF*) and the duration of the epidemic (*DUR*), obtained by the **BF** and **IS** models during the **NM** and **WM** experiments and some of those values available for the **KM** model, are reported in Table 4.9. The discrete distribution functions of *NIF* and *DUR* for the **WM** case are presented in Figure 4.4. We can observe on the table, that **IS** generates slightly larger values of *NIF* and *DUR*, but of the same order, than **BF**. This is an expected result when considering the differences between the models coefficients, in particular, the use (or not) of the proportions of infected animals which increases or decreases the probabilities of infection and detection. In fact, the main discrepancy is observed on the amplitude of the extreme scenarios (i.e., scenarios with numerous infected farms) which is higher for **IS** than **BF**. This can be observed in Figure 4.4, where the discrete densities are quite similar for both models except for the highest values (i.e., $NIF > 9$ and $DUR > 18$). This is confirmed, in the **WM** case, by the fact that the minimum, PI[2.5%], Q1, Q2 and Q3 values of both models are close and the PI[97.5%] and the maximum values are twice higher for **IS** than **BF**. However, in the **NM** experiments, since the mean value of *NIF* is higher (i.e., there are more extreme scenarios) than in **WM** ones, the difference between both models is amplified: Q2 and Q3 are also more than twice greater for **IS** than **BF**. When regarding the results produced by **KM** in the **WM** case, and taking into account that the considered region has a double number of farms and the area is smaller than Segovia, they can be considered as similar to those produced by **BF** and **IS**. Regarding the effect of applying or not the control measures, in both models we observe a similar behavior: the epidemic is reduced by ten when comparing the **NM** and **WM** experiments.

When considering the amplitude of the 1997-98 epidemic in Segovia, which consisted in 22 infected farms and had a duration of 60 days, it is difficult to compare it with the **BF** and **IS** results obtained in the **WM** case: the values of *DUR* are close to the real outbreak length, but the values of *NIF* are much lower. We have to take into account that 10 years separate the 2008 database used in our experiments and the real situation in Segovia in 1997. During this period, more than half of the farms have disappeared, due to an economic crisis in 2006 [184], and the control measures have been highly reinforced after the tremendous CSF epidemic in Europe during the years 1997 and 1998 [236]. Moreover, it is possible that this particular epidemic represents an extreme scenario of the model.

A good way to compare those real data with the **BF** and **IS** outputs, is to analyze the risk maps and see if the 1997-98 infected farms are in high risk zones.

The \bar{RI} risk maps generated by models **BF** and **IS** for the **NM** and **WM** experiments are presented in Figure 4.5. The Jenks Natural Breaks (JNB) classification, divided in 9 intervals corresponding to 9 gray colors (for a better understanding of the maps), is also reported in this Figure: the first three intervals $[0-0.03]$, $[0.03-0.05]$ and $[0.05-0.07]$ correspond to the low risk areas; the intervals $[0.07-0.10]$, $[0.10-0.12]$ and $[0.12-0.15]$ correspond to the medium risk areas; and the last three intervals $[0.15-0.17]$, $[0.17-0.20]$ and $[0.20-1]$ correspond to the high risk areas. This classification is obtained by considering the **NM** case (i.e., the worst case) with the **BF** model and is extended to the other maps. We point out that the JNB classifications obtained by **IS** are similar to the **BF** ones. As we can observe on those maps, the risk distributions obtained by both models decrease drastically from the **NM** cases to the **WM** ones. We can also see that, although both models identify similar high risk zones in the South-West of Segovia, **IS** concentrates the risk in some specific areas in the North and the East parts, whereas **BF** identifies the center of the region as presenting a high risk of CSFV spread. This is particularly visible on the **WM** maps.

Focusing on the **WM** case, we consider the farms infected during the 1997-98 epidemic in Segovia and see the risk zones where they are included. In Figure 4.6, we incorporate those farms to the **BF** and **IS** risk maps and we detail the area where most of the farms are included. We can see that, in the **BF** case, a large majority of the infected farms is situated in a dark gray (high risk) zone and the other farms in medium or low risk zones. In the **IS** case, the high risk zone does not include those farms which are mainly located in low risk areas. The mean \bar{RI} value of the 1998-97 infected farms given by the **BF** model is 0.201, which is included in the highest risk interval for the considered JNB classification. In the **IS** model, the mean risk value of those farms is 0.032, which corresponds to a low risk. This result tends to show that the maps generated by **BF** are more consistent with real data than those given by **IS**. This can be explained by the fact that our model uses the real commercial network between farms (i.e., transports of animals and SDA and INT groups), whereas this information is not suitably processed by **IS**. This shows the importance of using this database to obtain a fine representation of the risk areas, and one should include this input in an epidemiological model if it is available. As previously, we point out that 10 years separate the used databases from the 1997-98 outbreaks in Segovia and the real data considered for the **BF** and **IS** inputs, explaining why some farms are included in low risk zones, even in the **BF** map. However, this also shows the robustness of the **BF** risk maps, which seem to be valid for years different from those of the databases.

4.5 Analysis of the model robustness

In this Section, we are interested in verifying the robustness of our model regarding its inputs and outputs. To do so, in Section 4.5.1, we perform various experiments based on the Internal Validity and the sensitivity analysis of the model. The input data considered here are the ones of the Spanish province of Segovia presented in Section 4.4.2. In Section

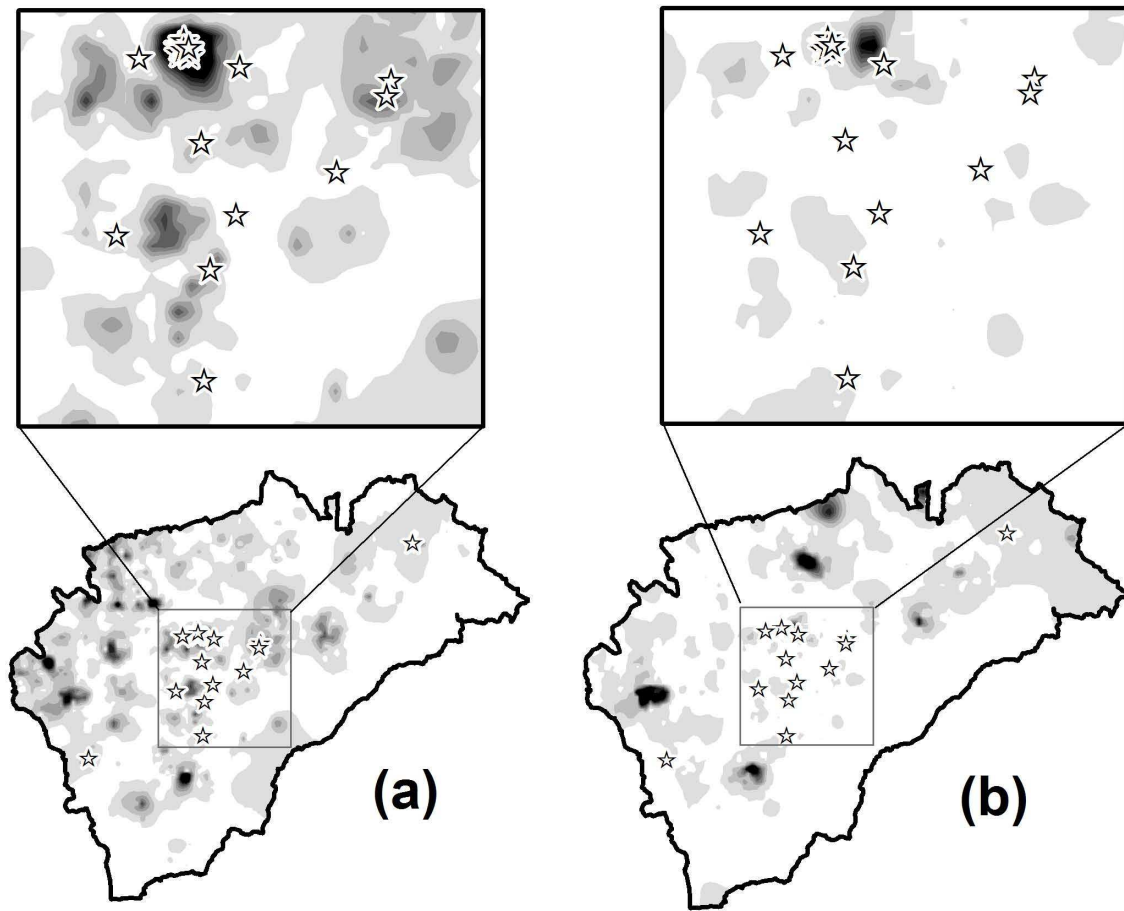


Figure 4.6: Normalized risk of CSFV introduction ($\bar{R}I$) interpolated maps obtained by models (a) **BF** (Be-FAST) and (b) **IS** (InterSpread Plus) for the **WM** (With Measures) experiment. We also report, with white stars (\star), the location of the farms infected during the 1997-98 CSF epidemic in Segovia. Furthermore, we present in the square region a zoom of the zone where most of those farms were situated (except two of them).

4.5.2, we present and discuss the obtained results.

4.5.1 Numerical experiments

a) Internal validity: Because the Be-FAST model is based on the combination of various stochastic processes, we were firstly interested in studying the variation of the output values from one execution to another. From a general point of view, as our model is based on a Monte-Carlo approach, a large number of scenarios (i.e., a high value for M) should be considered to ensure a good stability of the outputs between two different runs [136]. However, the larger the value of M the higher the resources and computational time required to obtain results, mainly if the model is complex. Thus, for the experiments presented here, we have considered an intermediate number of scenarios ($M = 1000$), which was considered to guaranty a certain stability of the outputs while requiring a reasonable computational time.

In order to check the robustness of the BE-FAST outputs, we ran 10 times the model keeping the same input values for the parameters as described in [185]. Then, for each output O_k , we compute its maximum ($MA(O_k)$), minimum ($MI(O_k)$) and mean ($M(O_k)$) values obtained during all those runs. We also have defined a mean error measure, ME, expressed in percentage respecting to $M(O_k)$ (which is taken as the reference value), by considering for all outputs, the following formula [130]:

$$ME(O_k) = \frac{1}{10} \sum_{i=1}^{10} \frac{|V_i(O_k) - M(O_k)|}{M(O_k)} \quad (4.5.1)$$

where $V_i(O_k)$ is the value of the output O_k obtained in the i^{th} run of the model.

b) Data validity: Information used to feed the model (i.e. farm type, geographical position, number of animals, ADS, integrator groups and pigs movements), which was described by [185], was provided by the Regional Government of Castile and Leon Region of Spain [182, 143] and was considered to be complete, updated and reliable. However, we intended to evaluate the impact that incomplete or not updated information regarding the farm demographics and characteristics (i.e. number and type of farms, incoming and outgoing movements of pigs and number of pigs per farm) has on model outcomes. To do so, we combined information regarding the farm demographics and characteristics from two different years (2005 and 2008) but keeping the number of farms similar to the 2008's one (by considering a Poisson Law). This approach was performed by randomly mixing 10% of the farms in 2005 and 90% of 2008 ten times to obtain ten different scenarios. At the end of those ten experiments, the output O_k was compared to its reference value $M(O_k)$ by considering the error formula (4.5.1). The objective was not only to assess the impact of using old information on model outcomes but also the impact that changes in farm demographics and characteristics has in the spread of CSFV.

c) Sensitivity analysis: The input values for the 33 parameters used to simulate the within- and between- farms transmission process and the detection and control of CSFV were obtained either from literature review or from expert opinion (Table 4.10). Whereas

ID	Parameter	Initial value	Reference
P1	Within-farm transmission parameter for farrowing pig farms, fattening pig farms and farrow-to-finish pig farms	$\beta_h = 8.52$; $\beta_h = 1.85$ and $\beta_h = 5.18$, respectively	Klinkenberg et al. (2002)
P2	Probability of infection by contact with vehicles transporting infected animals	Bernoulli [0.011]	Stegeman et al. (2002)
P3	Number of contacts with vehicles transporting products per farm at day t	Poisson [0.4]	Karsten et al. (2005a)
P4	Probability of infection by contacts with vehicles transporting products	Bernoulli [0.0068]	Stegeman et al. (2002)
P5	Number of contacts with people per farm at day t	Poisson [0.3]	Karsten et al. (2005a)
P6	Probability of infection by contact with people	Bernoulli [0.0065]	Stegeman et al. (2002)
P7	Probability of infection by local spread at day t	$\frac{I_{Ph}(t)}{N_h(t)} + \text{LSM}(h, k)$	Karsten et al. (2005b)
P8	Latent period (transition from infected to infectious state)	Poisson [7]	Karsten et al. (2005a)
P9	Transition from infectious to clinical signs state	Poisson [21]	Karsten et al. (2005a)
P10	Probability of detection based on clinical signs at day t before detection of the index case	Bernoulli [0.03]	Karsten et al. (2005b)
P11	Probability of detection based on clinical signs at day t after detection of the index case outside control and surv. zones.	Bernoulli [0.06]	Karsten et al. (2005b)
P12	Probability of detection based on clinical signs at day t after detection of the index case in the control zone	Bernoulli $\left[0.98 \left(\frac{I_{Ph}(t)}{N_h(t)}\right)\right]$	CyL expert opinion (2008)
P13	Probability of detection based on clinical signs at day t in the surveillance zone	Bernoulli $\left[0.95 \left(\frac{I_{Ph}(t)}{N_h(t)}\right)\right]$	CyL expert opinion (2008)
P14	Probability of detection based on serological test	Bernoulli [0.95]	MAPA (2006)
P15	Probability of restriction of animal movements on the detected as infected farm	Bernoulli [0.99]	CyL expert opinion (2008)
P16	Probability of restriction of vehicle movements on the detected as infected farm	Bernoulli [0.95]	CyL expert opinion (2008)
P17	Probability of restriction of people movements on the detected as infected farm	Bernoulli [0.80]	CyL expert opinion (2008)
P18	Probability of restriction of animal movements within the control and surveillance zone	Bernoulli [0.95]	CyL expert opinion (2008)
P19	Probability of restriction of vehicle movements within the control and surveillance zone	Bernoulli [0.90]	CyL expert opinion (2008)
P20	Probability of restriction of people movements within the control and surveillance zone	Bernoulli [0.70]	CyL expert opinion (2008)
P21	Probability of restriction of movements outside the control and surveillance zones	Bernoulli [0.4]	CyL expert opinion (2008)
P22	Probability to depopulate a detected as infected farm	Table [prob,day]: [0.11,0], [0.58,1], [0.2,2], [0.06,3], [0.04,4], [0.004,5], [0.003,6], [0.0015,7] and [0.0015,8]	Elbers et al. (1999)
P23	Maximum number of farms to be depopulated at day t	Poisson [20]	CyL expert opinion (2008)
P24	Time to repopulation of a depopulated farm	Poisson [90]	CyL expert opinion (2008)
P25	Maximum number of farms to be traced at day t	Poisson [60]	CyL expert opinion (2008)
P26	Probability of tracing an animal movement	Bernoulli [0.99]	CyL expert opinion (2008)
P27	Probability of tracing a vehicle/people movement	Bernoulli [0.70] and Bernoulli [0.40]	CyL expert opinion (2008)
P28	Duration (days) of general movement restriction	30	CyL expert opinion (2008)
P29	Number of farms visited by a person during one trip	3	CyL expert opinion (2008)
P30	Number of farms visited by a vehicle during one trip	4	CyL expert opinion (2008)
P31	Radius (km) applied for control and surveillance zones	3 km/10 km	CyL expert opinion (2008)
P32	Number of days for tracing	60	CyL expert opinion (2008)
P33	Duration (days) of control and surveillance zones	30/40	CyL expert opinion (2008)

Table 4.10: Parameters and values used in the Be-FAST model during the Sensitivity analysis experiments described in Section 4.5.1.

some of these inputs (from 1S to 11M and 22M) are well documented and used in other published models; other input values (from 12M to 33M, except 22M) are either not so well documented or based on potential subjective opinions (i.e. expert opinion). In any case, both (well documented or not) input values are likely to impact results and, therefore, should be carefully evaluated. In this section, we used sensitivity analysis (SA) to quantify the amount of change on outcomes when varying the input values used in the model. Specifically, we evaluate three aspects of the model: (1) the global behavior of the model when perturbing the whole set of parameters; (2) the impact of changes on each of the 33 parameters used in the model; and (3) the impact of deactivation of one infection route or one control measure (i.e. deactivation of group of parameters).

- **Sensitivity analysis of all parameters using a random perturbation of 10%:**
Firstly, we aim to study the model behavior when the whole set of parameters was

randomly perturbed. To do so, we ran the Be-FAST model perturbing randomly all the model parameters by a variation between $[-10\%, +10\%]$ of their reference value. This experiment was repeated ten times, and the output O_k compared to $ME(O_k)$ by considering the error formula (4.5.1).

- **Sensitivity analysis of the individual parameters used in the model:** In this section we intended to identify the most influential parameters in the model. This sensitivity analysis was performed by perturbing every single parameter $\pm 80\%$ their initial values. The values obtained for each O_k were compared with the ones obtained in the reference model by considering:

$$E(O_k) = \frac{|V(O_k) - M(O_k)|}{M(O_k)} \quad (4.5.2)$$

where $V(O_k)$ is the value of the output O_k obtained in the considered run of the model.

- **Sensitivity analysis of a set of parameters involved in the within- and between- farm transmission, detection and control processes of CSFV:** Here we evaluate the evolution of the epidemic when one of the infection routes or one of the control measures was neglected. Specifically we perform seven experiments: deactivation of transmission by local spread (i.e parameter **7S** set to 0), by animal transport (**2S** set to 0), by contact by persons (**5S**, **6S** and **29M** set to 0), by contact with vehicles (**3S**, **4S** and **30M** set to 0), and deactivation of measure of zoning (parameters **31M** and **33M** were set to 0), restriction of movements (**15M** to **21M** and **24M** set to 0); and tracing (**25M-28M** and **32M** to 0). In addition, we also evaluated the potential impact of not considering the within-farm transmission component, which was equivalent to consider a pure between-farm transmission model. This was done by considering all animals to be infected as soon as a farm becomes infected (parameter **1S** set to +). The obtained results were compared with their reference value using the error formula (4.5.2).

4.5.2 Results

a) Internal validity: The mean error $ME(O_k)$ value obtained for each considered output O_k after the ten runs of the model was about 3% (Table 4.11). The highest mean error was obtained for the proportion of infections due to people (5.5%) and the Risk value (5.2%). The range (max-min) for each output O_k was very small, with a mean value of 3.1% (Figure 4.10). The distribution of mean error for the Risk and R0 values was mainly concentrated in the areas with high pig density but, in general, was similar to the reference values with values for the Pearson correlation coefficient of $R^2=0.97$ and $R^2=0.99$ [243], respectively (Figure 4.8).

The internal validity allowed to verify the consistency of the stochastic model after different runs. The small mean error value obtained ($ME=3\%$) was found to be reasonably low considering the reduced number of Monte-Carlo simulations used [$M=1000$] and shows that this value of M gives a good ratio between computational complexity and output precision.

ID	Output	IV	DV	P10
O ₁	Number of infected farms	3.41	11.41	6.49
O ₂	Epidemic length (days)	1.66	2.37	3.56
O ₃	Infection due to local spread (%)	1.50	4.41	3.85
O ₄	Infection due to animal movements (%)	3.68	19.58	11.40
O ₅	Infection due to people contacts (%)	5.47	27.09	11.57
O ₆	Infection due to integrator group (%)	3.84	11.24	7.73
O ₇	Detection due to zoning (%)	2.53	18.16	10.84
O ₈	Detection due to clinical signs (%)	1.49	4.67	2.59
O ₉	Detection due to tracing (%)	2.59	19.91	5.11
O ₁₀	Number of farms affected by zoning	2.19	6.61	7.33
O ₁₁	Number of traced farms	3.95	16.70	5.57
O ₁₂	Mean R0 value (MR0)	2.62	17.9	3.07
O ₁₃	Mean Risk value (MRI)	5.16	16	5.90
Mean		3.08	13.54	6.54

Table 4.11: Mean relative error value (%), $ME(O_k)$ (see 4.5.1), of the outputs generated by performing the internal validity (IV), the data validity (DV) and when perturbing randomly up to $\pm 10\%$ the parameters of the Be-FAST model (P10).

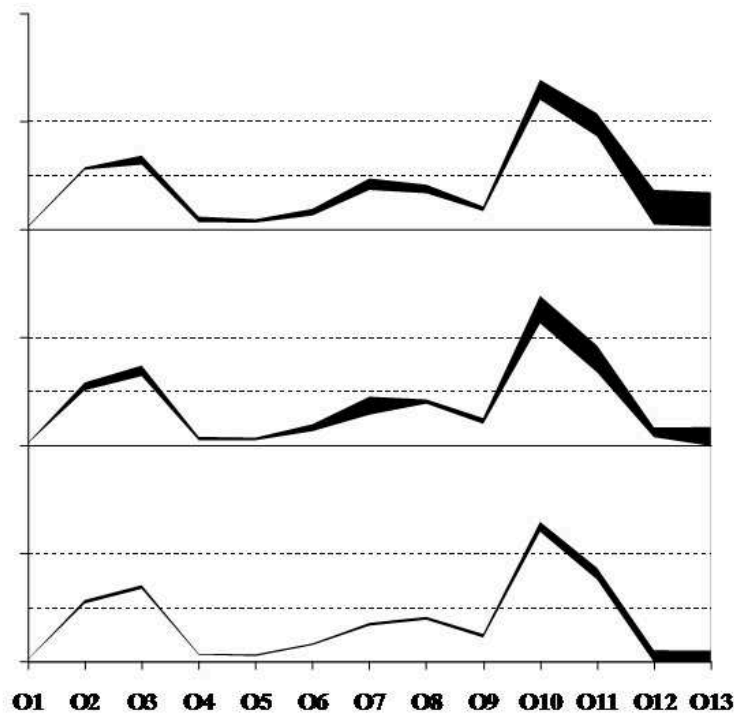


Figure 4.7: Range (min - max) for each output O_k obtained during (a) the internal validity, (b) the random perturbation of 10% of all parameters and (c) the data validity experiments.

b) Data validity: Results for the 10 experiments described in Section 2.3 are presented in Table 4.11. The mean error value was of 13.54%. The highest mean error value was found in the proportion of infections due to people (27.1%), the proportion of detections due to tracing (19.9%) and the proportion of infections due to animal movements (19.6%). The mean range (max-min) for each output O_k was of 11.5% (Figure 4.7), with the maximum ranges found in the mean R0 (Range=32%) and the mean Risk (Range=31%). The distribution of the Risk and R0 mean error values was only in part comparable to the reference values, with a $R^2 = 0.52$ and $R^2 = 0.48$, respectively (Figure 4.8).

As expected, the higher variation on model outputs was obtained after altering the input data used to feed the model (ME=13.54%) (Table 4.11). Moreover, the distribution of the areas at risk of introducing (High risk value) or spreading (High R0 value) the disease were importantly modified (almost 50%), compared with the reference scenario (Figure 4.7). These results highlight the importance of using updated and complete information regarding the area of study to obtain realistic and useful results for the decision making process.

c) Sensitivity analysis:

- **Sensitivity analysis of all parameters using a random perturbation of 10%:** All results are presented in Table 4.11. The mean error obtained after the perturbation of randomly selected parameters was 6.54%, which was of the order of the parameter perturbations (10%). The mean range (max-min) for each output O_k was of 10.3% (Figure 4.7), with the maximum ranges found in the number of farms affected by zoning (Range = 26%) and the number of traced farms (Range = 25%). The distribution of the mean error for the Risk and R0 values was not meaningfully different from the reference scenario ($R^2 = 0.94$ for both) (Figure 4.8).

The impact that variations (uncertainty) in the input values has on the model results was assessed by sensitivity analysis [243]. The low mean error (6.54%) obtained after the 10% random perturbation of all parameters confirmed the robustness of the model to general variations on the input parameters.

- **Sensitivity analysis of the individual parameters used in the model:** The most influencing parameters in the whole set of outputs were the probability of infection by local spread (**7S**), the time from infectious to clinical signs state (**9S**), the probability of detection based on clinical signs at day t after detection of the index case outside the control and surveillance zones (**11M**), the maximum number of farms to be depopulated at day t (**23M**), the probability of detection based on clinical signs at day t before detection of the index case (**10M**) and the latent period (**8S**) (Figure 4.9).

Specifically, the magnitude and duration of the CSFV epidemic was mostly impacted (>10% of change) by the probability of infection by local spread at day t (7S), the transition from infectious to clinical signs state (9S) and the probability of detection based on clinical signs at day t outside the control and surveillance zones (11M) (Figure 4.10). The maximum number of farms to be depopulated at day t (23M) was also very influential on the duration of the epidemic (Figure 4.10).

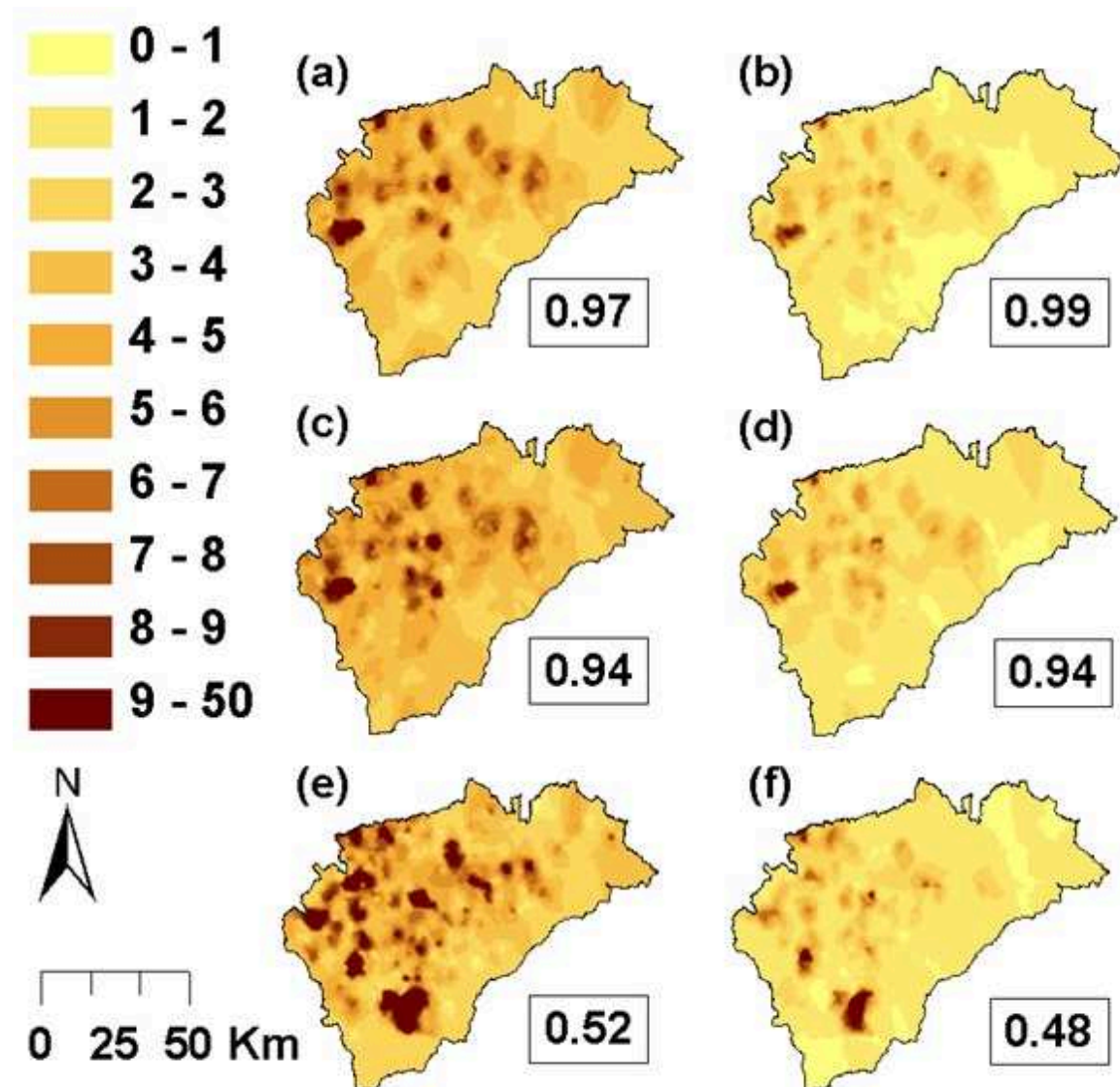


Figure 4.8: Spatial distribution of the mean error for the Risk (Left) and R0 (Right) values obtained during the experiments of (a,b) the internal validity, (c,d) the 10% random perturbation of input parameters and (e,f) the data validity, respectively. The Pearson correlation coefficient (R^2) between the reference R0 and Risk value and the values for each experiment is represented in the boxes.

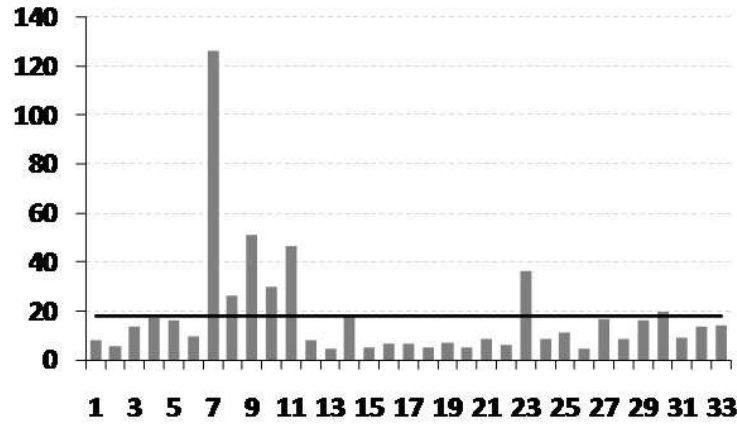


Figure 4.9: Mean error value of the outputs obtained by perturbing by $\pm 80\%$ of their initial value each input parameters of the Be-FAST model. The mean value considering the 33 experiments is represented by a black line.

The proportion of infections due to local spread, pig movements, people and other fomites were mostly sensitive to the number of contacts with vehicles transporting products per farm at day t (**3S**), probability of infection by contacts with vehicles transporting products (**4S**), the probability of infection by local spread at day t (**7S**) and the number of farms visited by a person (**29M**) and by a vehicle (**30M**) during one trip (Figure 4.10).

The proportion of detections by zoning, clinical signs and tracing were sensitive to the probability of infection by local spread at day t (**7S**), the transition from infectious to clinical signs state (**9S**), the probability of detection based on clinical signs after detection of the index case (**10M** and **11M**) and the number of farms visited by a person during one trip (**29M**) (Figure 4.11). The most important parameters regarding the number of traced farms and farms affected by zoning were the probability of infection by local spread at day t (**7S**), the transition from infectious to clinical signs state (**9S**), the number of farms visited by a vehicle during one trip (**30M**) and the radius (km) applied for control and surveillance zones (**31M**) (Figure 4.11). The mean Risk was sensitive mainly to the probability of infection by local spread at day t (**7S**) and the transition from infectious to clinical signs state (**9S**) (Figure 4.11).

The sensitivity analysis performed by doing a strong perturbation ($\pm 80\%$ the initial values) of the 33 input parameters intended to identify the most influential parameters in the model (Figure 4.9). As a result, the model results were found to be mostly sensitive ($>10\%$ of change) to six parameters: the probability of infection by local spread (**7S**), the time from infectious to clinical signs state (**9S**), the probability of detection based on clinical signs at day t after detection of the index case outside the control and surveillance zones (**11M**), the maximum number of farms to be depopulated at day t (**23M**), the probability of detection based on clinical signs at day t before detection of the index case (**10M**) and the latent

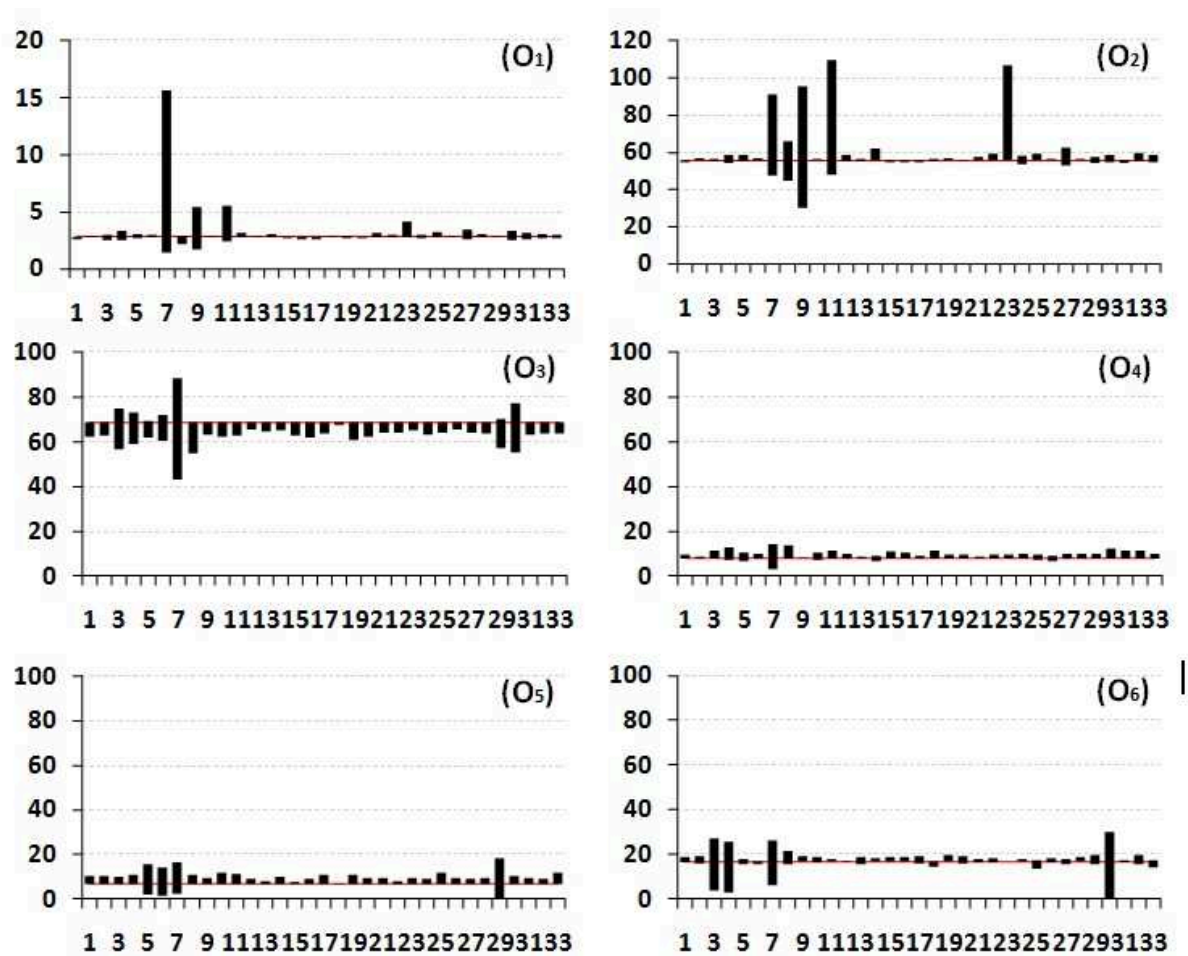


Figure 4.10: Bar plot representing the error range (min - max) for each outputs (O_1) - (O_6) obtained by perturbing $\pm 80\%$ each input parameters used in the Be-FAST model.

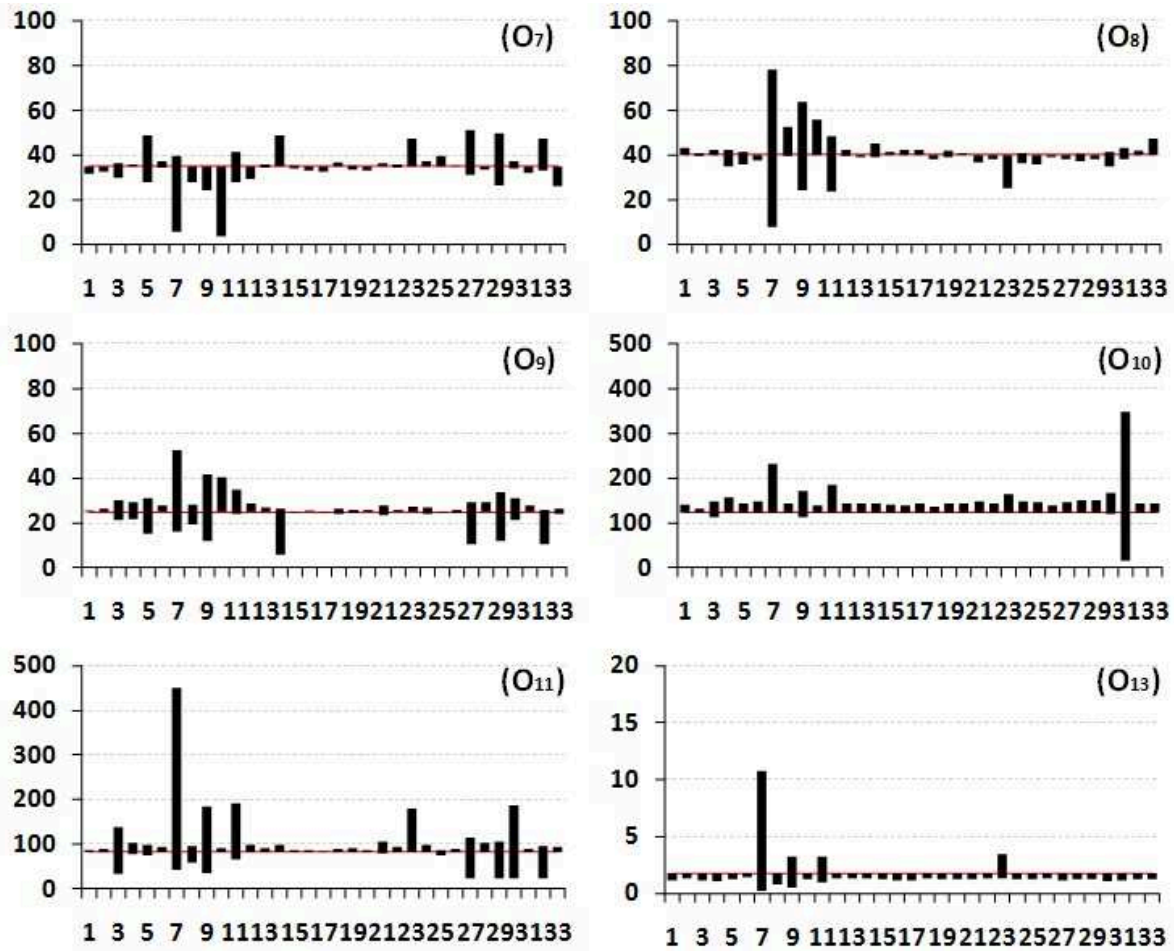


Figure 4.11: Bar plot representing the error range (min - max) for each outputs (O_7)-(O_{13}) obtained by perturbing $\pm 80\%$ each input parameters used in the Be-FAST model.

ID	AM	LS	PE	V	TC	ZO	RM	SIR
O ₁	7.17	51.04	7.17	9.94	20.66	15.89	10.51	100.47
O ₂	0.70	13.51	0.70	2.34	15.31	19.42	1.97	18.20
O ₃	4.42	^a	4.42	6.61	4.12	8.75	8.90	12.43
O ₄	^a	10.22	^a	22.84	1.61	39.08	30.74	21.99
O ₅	33.74	14.89	33.74	41.66	51.87	34.66	22.75	22.02
O ₆	4.69	10.62	4.69	^a	3.02	4.09	13.31	32.09
O ₇	1.47	91.15	1.47	1.75	24.61	^a	2.09	19.64
O ₈	5.44	97.43	5.44	4.42	21.38	28.95	9.13	49.63
O ₉	6.75	29.63	6.75	9.65	^a	18.60	17.75	52.85
O ₁₀	0.29	48.55	0.29	2.77	^a	26.07	28.16	108.98
O ₁₁	7.62	6.03	7.62	1.11	18.35	^a	20.07	25.91
O ₁₂	10.17	79.38	10.17	16.33	33.33	31.02	16.38	159.72
O ₁₃	11.17	79.61	11.17	15.36	32.63	30.56	16.42	158.88
Mean	7.87	43.38	9.67	11.46	20.37	22.56	15.08	58.23

^a Not applicable.

Table 4.12: Relative error values (%), $E(O_k)$ (4.5.2), of the outputs generated in the Be-FAST model by deactivating the transmission by animal movements (AM), by local spread (LS), by contact with people (PE) and by contact with vehicles (V); the measures of tracing (TC), zoning (ZO) and restriction of movements (RM); and the within-farm transmission process (SIR).

period (**8S**). Results are consistent with previous studies if we consider that those parameters are related with the local spread (**7S** and **23M**) and with the time from infection to detection of a CSFV-infected farm, also referred to as the high risk period, (**9S**, **11M**, **10M** and **8S**), which both has been traditionally identified as key-aspects to determine the magnitude and duration of an CSFV epidemic [140, 147]. Those results highlight the fact that studies that help to quantify the local spread and the time from infection to detection of CSFV in real epidemics, such as the one presented by [252, 251], are extremely useful to implement realistic estimates for disease spread models which ultimately will help to better prevent and control future CSFV epidemics.

- **Sensitivity analysis of a set of parameters involved in the within- and between- farm transmission or detection and control of CSFV:** The error values of the output generated by the Be-FAST model by deactivating one by one the ways of CSFV transmission or control measures are presented in Table 4.12.

The deactivation of the local spread transmission was the experiment that most affected the model outputs with a mean error of 43.38%. In particular, the magnitude and duration of the epidemic omitting this route reduces the number of infections and the epidemic length by 51.0% and 13.5%, respectively. Other ways of transmission such as animal movements or contacts with people or vehicles produced lower impact on the output with a mean error around 10%.

Similarly, zoning and tracing were the control measures that mostly impact model outcomes (ME \approx 22%), producing an increment in the magnitude and duration of the epidemic from 15% to 20%.

Finally, the suppression of the set of parameters involved in the within-farm transmission lead to the most important increase in the magnitude (100.5%) of the CSFV-

epidemic, with an important increase also in the epidemic duration (18.2%). In this case, the mean error on the output was about 58%.

The aim of the deactivation of a set of parameters involved in the spread, the detection and the control of CSFV was, firstly, to identify the most important routes of disease spread and, secondly, the most effective detection and control measures to be applied during a CSFV-epidemic. As a result, we found that local spread was the most important way of CSFV transmission in Segovia region. Also, the elimination of the within-farm transmission component was found to produce a much larger epidemic. This was an expected result as other influent parameters (such as **7S**) directly depend on the number of infected animals in the farm. Moreover, deactivation of the SIR component implies that all animals in a farm become infected immediately (at time 0) after the infection of the farm, which directly increases the probability of CSFV transmission by any route from this farm to any other farms. This result also reveals that the simplification of the model to a purely between-farm transmission model, will lead to an overestimation of the epidemic size and duration. In fact, the use of a combined within- and between- farm spread model will produce two times smaller epidemics than a simple between-farm transmission model. Policy makers should be aware of this potential overestimation of the simple between-farms spread models before interpreting and using the model for allocation of preventive and control measures.

On the other hand, tracing was considered the most effective measure to control the disease spread, because its suppression led to the major increase (+20%) in the magnitude and length of the epidemic. It is important to note that other the control measures were also important for the disease control, as their deactivation imply from 10% to 15% larger epidemics, but the role of tracing was crucial. This result reveals that the capabilities of the Animal Health Services to implement timing and effective tracing are extremely important to control disease spread in the CSFV-infected regions and may certainly determine the final sanitary and economical consequences of a CSFV-epidemic.

4.6 Conclusions and Perspectives

During this work, we first have presented an extended mathematical formulation of the spatial hybrid model called Be-FAST used for the study of CSFV spread into a region. In particular, we have described the processes considered to simulate the infection routes within and between farms and the control measures applied by authorities to fight this disease.

Then, in order to validate our model, we have performed various numerical experiments by considering farms and transports real databases of the Spanish province of Segovia. We have compared the results given by Be-FAST with those obtained by two other models (i.e., InterSpread Plus and the Individual-Based model presented in [147]) and with real outbreaks data from Spain and The Netherlands. We have seen that, due to some of the characteristics of Be-FAST (i.e., the combination of a Susceptible-Infected model with an Individual-Based model, the use of the proportions of infected animals in farms to

calibrate some model coefficients and the consideration of the real commercial network between farms), the outputs generated by our model fit better the real CSF epidemics information than the ones produced by InterSpread Plus. Those results tend to show the validity and the efficiency of our approach.

Finally, we have verified the robustness of Be-FAST regarding its inputs and outputs. To do so, extensive internal validity and sensitivity analysis processes have been performed. As a result, the model was found to be robust to general perturbations of input parameters, but sensitive to changes in the data used to feed the model and to elimination of the within-farm spread component. Parameters related with the local spread and with the time from infection to detection of disease were found to be the most influential in the model outputs. This reveals the need to incorporate good estimates for those parameters to get realistic results. Methods and results presented here may be useful for decision makers to better prevent and control future CSFV outbreaks.

One of the next steps of this work should be the incorporation in Be-FAST of a model to study the economic impact of a CSF epidemic in the considered region. It will be also interesting to use the risk map distribution to design CSF preventive campaigns, in order to reduce the economic losses and the risk of spread of future outbreaks. Those two last ideas are currently a work in progress and preliminary results can be found in [83]. Finally, as specified previously, other newsworthy studies will be the implementation of Be-FAST by using a fast programming language and to take into account the possible interactions with foreigner regions.

Chapter 5

Modeling and simulation in food engineering

Abstract - *In this Chapter, we focus on two problems related to the use of modeling and simulation techniques for studying the process of inactivation (i.e., folding process as defined in Section 3.1) of enzymes (i.e., proteins that act as catalysts of certain chemical reactions) in food:*

- *The first work, presented in Section 5.1, deals with the study of the effect of the combination of high pressure and thermal treatments on food processing, focusing on the inactivation of certain enzymes. Indeed, High Pressure (HP) Processing has turned out to be very effective in prolonging the shelf life of some food. The behavior and stability of the proposed models are checked by various numerical examples. Furthermore, various simplified versions of these models are presented and compared with each other in terms of accuracy and computational time. The models developed in this Section provide a useful tool to design suitable industrial equipments and optimize the processes.*
- *In the Section 5.2, we present an approach to reconstruct the dynamics of the protein folding process from the output of Monte Carlo simulations. Indeed, Proteins are often modeled at the molecular level as mechanical systems and their folding processes are simulated by using Monte Carlo methods. These techniques sample the configurational part of the protein's phase space and give spatial conformations and free energies. However, the velocities and derived kinetic properties remain unknown. To estimate those parameters, we assume that the folding process can be described as a diffusion process along the potential energy. Choosing suitable diffusivity profiles and using finite differences combined with the MSA, presented in Chapter 2, we match a specific kinetic property called relaxation rate between simulations and laboratory experiments. The results highlight the fact that the diffusive kinetics can be dependent on both the temperature and the potential energy.*

5.1 Modeling and simulation of High-Pressure/Temperature processes and inactivation of enzymes

5.1.1 Introduction

At present, the demand of safe and minimally processed food prepared for immediate consumption (ready-to-use and ready-to-eat) has increased significantly, in order to meet the needs of restaurants, collective dining rooms (colleges, companies, hospitals, residences, etc.) as well as domestic consumption.

One of the technologies that can be used for the preparation of these products is High Pressure (HP) processing, which has turned out to be very effective in prolonging the shelf life of some foods (cooked ham, juices, guacamole, oysters, etc.), being already a reality at an industrial level. One of the great advantages of this process is that they do not use additives, which consumers prefer to elude. Also, it avoids the use of high temperatures during the process (not like Pasteurization), which may have adverse effect on some nutritional properties of the food, its flavor, etc. (see, e.g., Ref. [115] and [128]).

This Section deals with the modelling and simulation of the effect of the combination of high pressure and thermal treatments on food processing, focussing on the inactivation of certain enzymes. Other studies for this kind of problems can be seen, for instance, in Ref. [64, 113] and [114]. Due to the high computational complexity needed to solve the full models (which include heat and mass transfer and non-constant thermophysical properties), we are also going to consider and to study some simplified versions of them. All of these models might become very important in order to design suitable industrial equipments and optimize the processes.

In Section 5.1.2 some models for enzymatic inactivation are presented. These models need the pressure and temperature profiles as an input. These quantities are obtained by means of the full and the simplified models developed in Section 5.1.3, where we also propose a sensitivity analysis. In Section 5.1.4 we couple those models in order to get numerical results for the distribution of temperature and inactivation of enzymes. Finally, in Section 5.1.5 we outline the final remarks, describing the steps proposed to optimize a thermal-HP process for any particular food and equipment.

5.1.2 Mathematical model for enzymatic inactivation

In order to predict the impact that HP-thermal processes have on the activity of some enzymes in food, we have implemented a particular first order kinetic model. Basically, it describes the evolution of the activity which depends on the pressure and the temperature, which are needed as an input for the model.

Several experimental protocols may be used to measure the enzymatic activity: typically, they take into account the variation of a suitable magnitude (for instance, oxygen concentration[129], optical density[173], carboxylgroups released from a pectin[176]) per time unit. According to the chosen magnitude, this activity is expressed in the corresponding units ($(\Delta \text{ppm of O}_2) \text{ s}^{-1}$, $\text{cm}^{-1} \text{ min}^{-1}$, $(\text{mL of } 0.01M \text{ NaOH}) \text{ min}^{-1}$ for the magnitudes expressed above, respectively).

The evolution of the activity A of an enzyme is often described by the following first-order kinetic equation[198, 253]:

$$\frac{dA}{dt}(t) = -\kappa(P(t), T(t)) A(t), \quad (5.1.1)$$

where t is the time (min), $P(t)$ is the pressure (MPa) at time t , $T(t)$ is the temperature (K) at time t , $\kappa(P, T)$ is the inactivation rate (min^{-1}) corresponding to the pressure-temperature conditions given by (P, T) and $A(t)$ is the activity of the enzyme under study.

In the literature, several mathematical formulae that describe $\kappa(P, T)$ can be found. They are usually based on equations that model the pressure-temperature dependence of chemical reactions and they are chosen depending on the enzyme being studied.

Equation (5.1.1) provides a macro behaviour of the enzymatic activity. Nevertheless, it is not capable of capturing the non-homogeneous activity distribution that could appear (due to non-homogeneous temperature distribution) inside the food. Here we restrict the exposition to two equations that describe some types of enzymatic inactivation, which are then going to be used for numerical simulations in Section 5.1.4:

- The first equation is provided by a suitable combination of Arrhenius equation (modelling the temperature dependence) and Eyring equation (modelling the pressure dependence)[173]:

$$\kappa(P, T) = \kappa_{\text{ref}} \exp \left(-B \left(\frac{1}{T} - \frac{1}{T_{\text{ref}}} \right) \right) \exp (-C(P - P_{\text{ref}})), \quad (5.1.2)$$

where T_{ref} is a reference temperature (K), P_{ref} is a reference pressure (MPa), κ_{ref} is the inactivation rate at reference conditions (min^{-1}), B is the parameter that expresses the temperature dependence of κ (K) and C is the parameter that expresses the pressure dependence of κ (MPa^{-1}).

- The second equation is obtained by considering Eyring's transition state theory adapted to enzymatic study[129, 176]:

$$\begin{aligned} \kappa(P, T) = & \kappa_{\text{ref}} \exp \left(\frac{-\Delta V_{\text{ref}}}{RT} (P - P_{\text{ref}}) \right) \exp \left(\frac{\Delta S_{\text{ref}}}{RT} (T - T_{\text{ref}}) \right) \\ & \exp \left(\frac{\Delta \nu}{2RT} (P - P_{\text{ref}})^2 \right) \exp \left(\frac{-2\Delta \zeta}{RT} (P - P_{\text{ref}})(T - T_{\text{ref}}) \right) \\ & \exp \left(\frac{\Delta C_p}{RT} \left(T \ln \frac{T}{T_{\text{ref}}} - 1 \right) + T_{\text{ref}} \right) + \text{high order terms}, \end{aligned} \quad (5.1.3)$$

where $R = 8.314$ ($\text{J mol}^{-1}\text{K}^{-1}$) is the universal gas constant, ΔV_{ref} is the volume change at reference conditions ($\text{cm}^3\text{mol}^{-1}$), ΔS_{ref} is the entropy change at reference conditions ($\text{J mol}^{-1}\text{K}^{-1}$), ΔC_p is the heat capacity change ($\text{J mol}^{-1}\text{K}^{-1}$), $\Delta \zeta$ is the thermal expansibility factor ($\text{cm}^3\text{mol}^{-1}\text{K}^{-1}$) and $\Delta \nu$ is the compressibility factor ($\text{cm}^6\text{J}^{-1}\text{mol}^{-1}$). Depending on the enzyme that is being studied, higher order terms can be added to (5.1.3) in order to refine the approximation of the pressure-temperature dependence of the activity[176].

The parameters of the selected equation have been estimated using regression techniques on the data provided by experimental measurements of the activity[69]. We point out that they may depend on the enzymatic production lots used in the experiment (see, for instance, Ref. [173]).

Once the equation and parameters of κ have been obtained, the solution at time t of (5.1.1) is obviously given by

$$A(t) = A(0) \exp \left(- \int_0^t \kappa(P(\sigma), T(\sigma)) d\sigma \right). \quad (5.1.4)$$

Putting this into practice, we set $A(0) = 100$ units, which is equivalent to assume that A is the percent value of the initial activity.

These models have been successfully applied when studying the inactivation of various enzymes with different conditions of pressure and temperature (see Ref. [129, 173] and [176]). However, they can be used only for cases where the temperature is known (typically by experimental measurements). Therefore, these models do not allow to perform numerical optimization in general situations without temperature evolution data. Furthermore, unless we know the distribution of the temperature and its evolution inside the food, this model is only able to describe macro values for the enzymatic activity without being able to provide the possible non-homogeneous distribution of the activity inside the food. All these drawbacks will be overcome in the following sections by developing models (Section 5.1.3) that are capable of describing the distribution of temperature inside the food sample and by coupling them (Section 5.1.4) with the models presented in this section.

5.1.3 Heat and Mass Transfer Modelling

When HP is applied in Food Technology, it is necessary to take into account the thermal effects that are produced by variations of temperature due to the compression/expansion that takes place in both the food sample and the pressurizing fluid.

During and after the compression, there is heat exchange between the pressure chamber, the pressure medium and the food sample. As a result, we get a time-dependent distribution of temperatures. In the fluid medium (the pressurizing fluid and also the food sample when it is in liquid state) temperatures variation implies fluid density variation, leading to free convection during the high pressure process. Therefore, conduction and convection have been considered in these models, taking into account heat and mass transfer[63, 159].

Often, HP experiments are carried out in a cylindrical pressure vessel (typically a hollow steel cylinder) previously filled with the packed food and the pressurizing fluid. The user may choose if the food sample is going to be cooled or warmed during the process.

Due to the characteristics of this kind of processes, we assume that thermally induced flow instabilities are negligible. Therefore, axial symmetry allows us to use cylindrical coordinates and the corresponding domain given by half a cross section (intersection of the cylinder with a plane containing the axis). Let us consider four two-dimensional sub-domains (see Figure 5.1):

- Ω_F : domain where the food sample is located.
- Ω_C : cap of the sample holder (typically a rubber cap).
- Ω_P : domain occupied by the pressurizing medium.
- Ω_S : domain of the steel that surrounds the domains mentioned above.

Then, our domain in the (r, z) -coordinates is the rectangle $\Omega = [0, L] \times [0, H]$ defined by $\overline{\Omega} = \overline{\Omega_F \cup \Omega_C \cup \Omega_P \cup \Omega_S}$, where $\{0\} \times (0, H)$ generates the axis of symmetry.

On the boundary of Ω , which is denoted by Γ , we distinguish:

- $\Gamma_r \subset \{L\} \times (0, H)$, where the temperature will be known.
- $\Gamma_{up} = [0, L] \times \{H\}$, where heat transfer with the room where the equipment is located may take place.
- $\Gamma \setminus \{\Gamma_r \cup \Gamma_{up}\}$, with zero heat flux, either by axial symmetry or by isolation of the equipment.

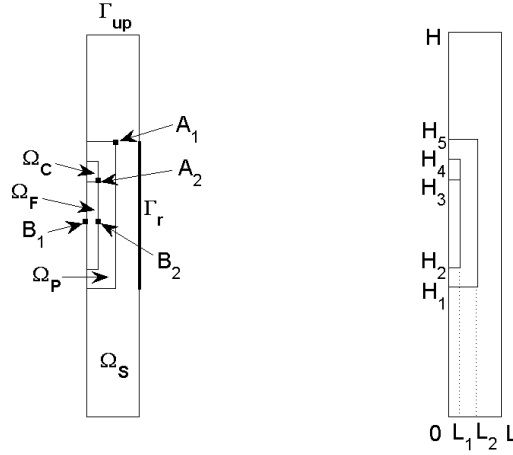


Figure 5.1: Computational domain.

We denote by Ω^* , Ω_F^* , Ω_C^* , Ω_P^* , Ω_S^* , Γ^* , Γ_r^* and Γ_{up}^* the domains generated when rotating Ω , Ω_F , Ω_C , Ω_P , Ω_S , $\Gamma \setminus (\{0\} \times (0, H))$, Γ_r and Γ_{up} along the axis of symmetry (in the 3D space), respectively.

For the mathematical model we distinguish two significant cases: solid and liquid type foods (see Sections 5.1.3 and 5.1.3, respectively).

Solid type foods

Heat transfer by conduction When solid type foods are considered, the starting point is the heat conduction equation for the temperature T (K)

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = \alpha \frac{dP}{dt} T \quad \text{in } \Omega^* \times (0, t_f), \quad (5.1.5)$$

where $\rho = \rho(T, P)$ is the density (Kg m^{-3}), $C_p = C_p(T, P)$ is the heat capacity ($\text{J Kg}^{-1}\text{K}$), $k = k(T, P)$ is the thermal conductivity ($\text{W m}^{-1}\text{K}^{-1}$) and t_f is the final time (s). The right hand side of the equation denotes the internal heat generation due to pressure change (see Ref. [219]). Here $P = P(t)$ is the pressure (Pa) applied by the equipment (this is chosen by the user within the machine limitations) and $\alpha = \alpha(T, P)$ is given by

$$\alpha = \begin{cases} \text{thermal expansion coefficient (K}^{-1}\text{) of the food in } \Omega_F^*, \\ \text{thermal expansion coefficient (K}^{-1}\text{) of the pressurizing fluid in } \Omega_P^*, \\ 0, \text{ elsewhere.} \end{cases}$$

This term results from the following law:

$$\frac{\Delta T}{\Delta P} = \frac{\alpha TV}{MC_p} = \frac{\alpha T}{\rho C_p},$$

where ΔT denotes the temperature change due to the pressure change ΔP , V (m^3) is the volume and M (Kg) is the mass.

The conductive heat transfer equation (5.1.5) is completed with appropriate initial and boundary conditions depending on the HP machine, in order to determine the solution that we are looking for. Here, we have used the same conditions as in Ref. [219] for a pilot unit (ACB GEC Alsthom, Nantes, France) located at the Instituto del Frío, CSIC, Spain:

$$\begin{cases} k \frac{\partial T}{\partial \mathbf{n}} = 0 & \text{on } \Gamma^* \setminus (\Gamma_r^* \cup \Gamma_{up}^*) \times (0, t_f), \\ k \frac{\partial T}{\partial \mathbf{n}} = h(T_{\text{amb}} - T) & \text{on } \Gamma_{up}^* \times (0, t_f), \\ T = T_r & \text{on } \Gamma_r^* \times (0, t_f), \\ T(0) = T_0 & \text{in } \Omega^*, \end{cases} \quad (5.1.6)$$

where \mathbf{n} is the outward unit normal vector on the boundary of the domain, T_0 is the initial temperature, T_r is the refrigeration or heating temperature that stays constant in Γ_r^* (cooling or warming the food sample), T_{amb} is the (constant) ambient temperature and h ($\text{W m}^{-2}\text{K}^{-1}$) is the heat transfer coefficient.

By using cylindrical coordinates and axial symmetry, system (5.1.5)–(5.1.6) may be rewritten as the following 2D problem:

$$\begin{cases} \rho C_p \frac{\partial T}{\partial t} - \frac{1}{r} \frac{\partial}{\partial r} \left(r k \frac{\partial T}{\partial r} \right) - \frac{\partial}{\partial z} \left(k \frac{\partial T}{\partial z} \right) = \alpha \frac{dP}{dt} T & \text{in } \Omega \times (0, t_f), \\ k \frac{\partial T}{\partial \mathbf{n}} = 0 & \text{on } \Gamma \setminus (\Gamma_r \cup \Gamma_{up}) \times (0, t_f), \\ k \frac{\partial T}{\partial \mathbf{n}} = h(T_{\text{amb}} - T) & \text{on } \Gamma_{up} \times (0, t_f), \\ T = T_r & \text{on } \Gamma_r \times (0, t_f), \\ T(0) = T_0 & \text{in } \Omega. \end{cases} \quad (5.1.7)$$

This model is suitable when the filling ratio of the food sample inside the vessel is much higher than the one of the pressurizing medium. This has been shown in Ref. [219], where the model has been validated with several comparisons between numerical and experimental results. In Ref. [219] they also show that when the filling ratio of the food inside the vessel is not much higher than the one of the pressure medium, the solution of this model is very different from the experimental measurements. Two ways of solving that inconvenience are the following:

- To use the same model but with an apparent conductivity value for the pressurizing medium higher than the real one. This method will not give good results for the temperature distributions in the pressurizing fluid but can give acceptable results inside the food. We are not going to discuss this possibility in this Section.
- As shown in Ref. [219], the model can be improved by including the convection phenomenon that takes place in the pressurizing medium. The resulting model is more expensive from a computational point of view but the results are more accurate. We discuss this methodology in Section 5.1.3.

Heat transfer by conduction and convection The non-homogeneous temperature distribution induces a non-homogeneous density distribution in the pressurizing medium and consequently a buoyancy fluid motion. In other words, free convection.

This fluid motion may strongly influence the temperature distribution. In order to take into account this fact, a non-isothermal flow model is considered. Therefore, we suppose that the fluid velocity field, \mathbf{u} (m s^{-1}), satisfies the Navier–Stokes equations for compressible Newtonian fluid under Stoke’s assumption (see, for instance, Ref. [10]). The resulting system of equations is:

$$\left\{ \begin{array}{l} \rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) + \rho C_p \mathbf{u} \cdot \nabla T = \alpha \frac{dP}{dt} T \quad \text{in } \Omega^* \times (0, t_f), \\ \rho \frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot \eta (\nabla \mathbf{u} + \nabla \mathbf{u}^t) + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} \\ \quad = -\nabla p - \frac{2}{3} \nabla (\eta \nabla \cdot \mathbf{u}) + \rho \mathbf{g} \quad \text{in } \Omega_p^* \times (0, t_f), \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad \text{in } \Omega_p^* \times (0, t_f), \end{array} \right. \quad (5.1.8)$$

where \mathbf{g} is the gravity vector (m s^{-2}), $\eta = \eta(T, P)$ is the dynamic viscosity (Pa s), $p = p(x, t)$ is the pressure generated by the mass transfer inside the fluid, and $P + p$ is the total pressure (Pa) in the pressurizing medium Ω_p^* .

We point out that on the right hand side of the first equation of (5.1.8) we could have written $\alpha \frac{d(P+p)}{dt} T$, but we have supposed that the internal heat generation due to mass transfer is negligible. On the right hand side of the second equation of (5.1.8) we have written ∇p , since $P = P(t)$ depends only on time and therefore $\nabla(P + p) = \nabla p$. As in the previous section, the density $\rho = \rho(T, P)$ is a known state function.

System (5.1.8) has been completed with appropriate point, boundary and initial conditions:

$$\left\{ \begin{array}{ll} k \frac{\partial T}{\partial \mathbf{n}} = 0 & \text{on } \Gamma^* \setminus (\Gamma_r^* \cup \Gamma_{up}^*) \times (0, t_f), \\ k \frac{\partial T}{\partial \mathbf{n}} = h(T_{\text{amb}} - T) & \text{on } \Gamma_{up}^* \times (0, t_f), \\ T = T_r & \text{on } \Gamma_r^* \times (0, t_f), \\ \mathbf{u} = 0 & \text{on } \Gamma_P^* \times (0, t_f), \\ T(0) = T_0 & \text{in } \Omega^*, \\ p = 10^5 & \text{in } \mathbf{A}_1 \times (0, t_f), \end{array} \right. \quad (5.1.9)$$

where \mathbf{A}_1 is a corner point of Γ_P^* , which is the boundary of Ω_P^* .

We remark that the point condition at \mathbf{A}_1 means that the total pressure ($P + p$) at this point is equal to the equipment pressure P plus the atmospherical pressure.

As shown in Section 5.1.3 for the conductive heat transfer model (see system (5.1.7)), system (5.1.8)–(5.1.9) can be also rewritten as an equivalent 2D problem by using cylindrical coordinates (we do not write here the resulting system). All the numerical experiments that have been considered in this Section were carried out on the 2D version of the corresponding equations.

Liquid type foods

For liquid type foods we propose a model considering that convection also occurs in the region Ω_F and we distinguish two separated velocity fields, \mathbf{u}_F and \mathbf{u}_P , for the food and the pressurizing fluid, respectively. We point out that the pressurizing medium and the food are separated by the sample holder and do not mix. As in Section 5.1.3, we assume that the pressurizing fluid is compressible and Newtonian.

Note that in this case the convection plays again an important role, not only in the pressurizing fluid, but also in the liquid type food sample. Therefore, the neglect of its effect (as done in the solid case), would produce results that would be very different to the real thermal behaviour. A detailed dimensional analysis for equations involving convection phenomena during high pressure treatment of liquid media can be found in Ref. [159].

Assuming that the food is a compressible Newtonian fluid, the governing equations

are

$$\left\{ \begin{array}{ll} \rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) + \rho C_p \mathbf{u} \cdot \nabla T = \alpha \frac{dP}{dt} T & \text{in } \Omega^* \times (0, t_f), \\ \rho \frac{\partial \mathbf{u}_F}{\partial t} - \nabla \cdot \eta (\nabla \mathbf{u}_F + \nabla \mathbf{u}_F^t) + \rho (\mathbf{u}_F \cdot \nabla) \mathbf{u}_F \\ \quad = -\nabla p - \frac{2}{3} \nabla (\eta \nabla \cdot \mathbf{u}_F) + \rho \mathbf{g} & \text{in } \Omega_F^* \times (0, t_f), \\ \rho \frac{\partial \mathbf{u}_P}{\partial t} - \nabla \cdot \eta (\nabla \mathbf{u}_P + \nabla \mathbf{u}_P^t) + \rho (\mathbf{u}_P \cdot \nabla) \mathbf{u}_P \\ \quad = -\nabla p - \frac{2}{3} \nabla (\eta \nabla \cdot \mathbf{u}_P) + \rho \mathbf{g} & \text{in } \Omega_P^* \times (0, t_f), \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}_F) = 0 & \text{in } \Omega_F^* \times (0, t_f), \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}_P) = 0 & \text{in } \Omega_P^* \times (0, t_f), \end{array} \right. \quad (5.1.10)$$

with point, boundary and initial conditions:

$$\left\{ \begin{array}{ll} k \frac{\partial T}{\partial \mathbf{n}} = 0 & \text{on } \Gamma^* \setminus (\Gamma_r^* \cup \Gamma_{up}^*) \times (0, t_f), \\ k \frac{\partial T}{\partial \mathbf{n}} = h(T_{\text{amb}} - T) & \text{on } \Gamma_{up}^* \times (0, t_f), \\ T = T_r & \text{on } \Gamma_r^* \times (0, t_f), \\ \mathbf{u}_F = 0 & \text{on } \Gamma_F^* \times (0, t_f), \\ \mathbf{u}_P = 0 & \text{on } \Gamma_P^* \times (0, t_f), \\ T = T_0 & \text{in } \Omega^*, \\ p = 10^5 & \text{in } \mathbf{A}_1 \times (0, t_f), \\ p = 10^5 & \text{in } \mathbf{A}_2 \times (0, t_f), \end{array} \right. \quad (5.1.11)$$

where Γ_F^* denotes the boundary of Ω_F^* and $\mathbf{A}_1, \mathbf{A}_2$ are corner points of Γ_P^* and Γ_F^* , respectively (see Figure 5.1). As in the previous sections, we will use a 2D-axially symmetric version of (5.1.10)–(5.1.11).

We point out that when the food is a non-Newtonian fluid, the second equation of system (5.1.10) should be replaced by the corresponding momentum balance equation. This complicates considerably the model and will not be considered here.

Full models considered

For practical purposes, in the following sections we are going to focus only on two relevant situations:

- Solid type food with a big filling ratio. According to the results of Ref. [219] discussed at the end of Section 5.1.3, we will consider system (5.1.7) as the corresponding full model (denoted by SFull).
- Liquid type food with a small filling ratio. In this case, equations (5.1.10)–(5.1.11) will be considered as the full model (denoted by LFull).

Model sensitivity

In practice, the coefficients used in equations (5.1.7)–(5.1.11) are usually approximated using experimental data with a standard deviation lower than $\pm 5\%$ of the value [255]. Furthermore, due to equipment limitations, some experimental discrepancies could occur during the process (for instance, the pressure curve could be not strictly respected, measure errors in initial temperature could occur, etc). In order to study the impact of those errors on the temperature and enzymatic activity evolutions during the process, we perform a sensitivity study on the considered models.

More precisely, we generate $N \in \mathbb{N}$ perturbed models (with corresponding temperature solutions T_{per} , $\text{per} = 1, 2, \dots, N$) from the original one, with ρ , C_p , k , α , η , T_0 , T_r and P perturbed randomly by $\pm 5\%$. To perform the sensitivity analysis, for an arbitrary function $f : D \rightarrow \mathbb{R}$ with $D \subset \mathbb{R}^p$, we define the mean value

$$\mathcal{M}(f; D) = \frac{1}{|D|} \int_D f(z) dz, \quad (5.1.12)$$

where $|D|$ is the measure of D . Then, we compute the average of the mean temperature errors

$$\text{AET} = \frac{1}{N} \sum_{\text{per}=1}^N \mathcal{M}(|T - T_{\text{per}}|; \Omega \times (0, t_f)), \quad (5.1.13)$$

that will be compared with the mean temperature $\mathcal{M}(T; \Omega \times (0, t_f))$, in order to obtain the corresponding percent relative error

$$\text{RAET} = 100 \times \frac{\text{AET}}{\mathcal{M}(|T|; \Omega \times (0, t_f))}. \quad (5.1.14)$$

Since the enzymatic activity model studied in Section 5.1.2 needs as an input the temperature only of the food sample, we highlight the behaviour of the models when focusing on the domain Ω_F . To do that, we define the average temperature error in the food sample, AET_F , and the corresponding percent relative error, RAET_F , by changing Ω by Ω_F in (5.1.13) and (5.1.14), respectively.

Simplified models

In order to reduce the computational complexity needed to solve the full models (5.1.7) and (5.1.10)–(5.1.11), it is interesting to consider some simplified versions (called ‘simplified models’), that are cheaper to evaluate and with results that do not vary too much from those of the full models. Indeed, simplified models are useful when they are used, for example, during optimization processes that need to solve the models many times for different data [135, 137, 231].

We carry out the study of the numerical characteristics of one simplified version of the solid type food model (5.1.7) and two simplified versions of the liquid type food model (5.1.10)–(5.1.11) described previously.

For the solid type food we consider a simplified model with constant coefficients, by setting C_p , k , α , ρ and η to their mean value (\bar{C}_p , \bar{k} , $\bar{\alpha}$, $\bar{\rho}$ and $\bar{\eta}$, respectively) in the range of temperature and pressure considered in the process. This model is denoted by SCC.

On the other hand, for the liquid type food we consider a first simplified model with constant coefficients, as in the SCC model, except for the density ρ , which remains dependent on temperature and pressure (in order to keep the effect of the gravitational forces). This model is denoted by LCC.

The second simplified model for the liquid case is based on the Boussinesq approximation. More precisely, the coefficients C_p , k , α and η are considered to be constant as in the LCC model; ρ is also chosen as a constant value $\bar{\rho}$, except for the gravitational force $\rho \mathbf{g}$ that appears in the second and third equation of the system (5.1.10). Furthermore, the food and the pressurizing fluids are assumed to be incompressible. This model is denoted by LB and given by

$$\left\{ \begin{array}{ll} \bar{\rho} \bar{C}_p \frac{\partial T}{\partial t} - \bar{k} \nabla^2 T + \bar{\rho} \bar{C}_p \mathbf{u} \cdot \nabla T = \bar{\alpha} \frac{dP}{dt} T & \text{in } \Omega^* \times (0, t_f), \\ \bar{\rho} \frac{\partial \mathbf{u}_F}{\partial t} - \bar{\eta} \nabla^2 \mathbf{u}_F + \bar{\rho} (\mathbf{u}_F \cdot \nabla) \mathbf{u}_F = -\nabla p + \rho \mathbf{g} & \text{in } \Omega_F^* \times (0, t_f), \\ \bar{\rho} \frac{\partial \mathbf{u}_P}{\partial t} - \bar{\eta} \nabla^2 \mathbf{u}_P + \bar{\rho} (\mathbf{u}_P \cdot \nabla) \mathbf{u}_P = -\nabla p + \rho \mathbf{g} & \text{in } \Omega_P^* \times (0, t_f), \\ \nabla \cdot \mathbf{u}_F = 0 & \text{in } \Omega_F^* \times (0, t_f), \\ \nabla \cdot \mathbf{u}_P = 0 & \text{in } \Omega_P^* \times (0, t_f), \end{array} \right. \quad (5.1.15)$$

with boundary and initial conditions given by (5.1.11).

We are interested in evaluating the efficiency of the simplified models. With this aim, we compare the simplified model SCC with the full model SFull and the simplified models LCC and LB with the full model LFull. More precisely, denoting by T_{sim} the temperatures obtained by solving each of them, we compute the mean error committed in the whole domain,

$$\text{ET} = \mathcal{M}(|T - T_{\text{sim}}|; \Omega \times (0, t_f)), \quad (5.1.16)$$

and the corresponding percent relative error

$$\text{RET} = 100 \times \frac{\text{ET}}{\mathcal{M}(|T|; \Omega \times (0, t_f))}. \quad (5.1.17)$$

As in Section 5.1.3, we emphasize the behaviour of the models focusing on the domain Ω_F . To do that, we define ET_F , and RET_F , by taking Ω_F instead of Ω in (5.1.16) and (5.1.17), respectively.

Numerical tests

For the numerical experiments we have considered the size of the pilot unit (ACB GEC Alsthom, Nantes, France) that was used in Ref. [219]. More precisely, $L = 0.09$ m, $H = 0.654$ m, $L_2 = 0.05$ m, $H_1 = 0.222$ and $H_5 = 0.472$ m (see Figure 5.1).

We work with the two representative examples described in Section 5.1.3. The size and location of the sample and the rubber cap are given by $H_3 = 0.404$ m and $H_4 = 0.439$ in both cases; $L_1 = 0.045$ m and $H_2 = H_1$ in the solid case, and $L_1 = 0.02$ m and $H_2 = 0.294$ m in the liquid case (see Figure 5.1).

We present numerical tests computed in cylindrical coordinates using the Finite Element Method (FEM) solver COMSOL Multiphysics 3.4. Velocity and pressure spatial discretization is based on P2–P1 Lagrange Finite Elements satisfying the Ladyzhenskaya, Babuska and Brezzi (LBB) stability condition. The time integration is performed using the Variable–Step–Variable–Order (VSVO) BDF–based strategy implemented in this platform. The nonlinear systems are solved with a damped Newton method. The algebraic linear systems are solved using UMFPACK (Unsymmetric MultiFrontal method for sparse linear systems) combined with the stabilization technique GLS (Galerkin Least Squares). All computations for the numerical tests have been performed on a Quad–Core processor with 3.4 GHz/Core and 8Gb of RAM.

The physical parameters of the pressurizing medium and the liquid type food are supposed to be equal to those of water and dependent on temperature and pressure. More precisely, ρ , C_p and k are computed through a *shifting approach*[218] from atmospheric pressure. For the parameter α we use the expression described in Ref. [217]. Finally, dynamic viscosity η is computed by a piecewise linear interpolation, with data obtained from Ref. [166].

On the other hand, we have chosen tylose as an example of solid type food. This gel has similar properties to meat[213]. The corresponding coefficients are obtained from Ref. [218] for atmospheric pressure. A rescaling procedure[114, 213] and a piecewise linear interpolation have been applied for other values of pressure.

For general cases, where the thermophysical properties of a particular food are unknown, mathematical tools for inverse problems may be needed to identify these parameters. For example, in Ref. [108] the authors discuss how to identify the heat transfer coefficient for a particular prototype. Identification of coefficients depending on temperature is considered in Ref. [93], in a rigorous mathematical way, for a general abstract framework.

In both cases, liquid and solid type food, the thermophysical properties of the steel and the rubber cap of the sample holder are assumed to be constant. More precisely, $\rho = 7833 \text{ Kg m}^{-3}$, $C_p = 465 \text{ J Kg}^{-1}\text{K}^{-1}$ and $k = 55 \text{ W m}^{-1}\text{K}^{-1}$ for steel and $\rho = 1110 \text{ Kg m}^{-3}$, $C_p = 1884 \text{ J Kg}^{-1}\text{K}^{-1}$ and $k = 0.173 \text{ W m}^{-1}\text{K}^{-1}$ for rubber are token.

The ambient temperature, the reference temperature and the heat transfer coefficient used in the tests are $T_{\text{amb}} = 19.3^\circ\text{C}$, $T_r = 40^\circ\text{C}$ and $h = 28 \text{ W m}^{-2}\text{K}^{-1}$, respectively.

For each type of food, we consider two high pressure processes with different initial temperature and pressure curve:

i) Process P1: The initial temperature is

$$T_0 = \begin{cases} 40^\circ\text{C} & \text{in } \Omega_S, \\ 22^\circ\text{C} & \text{in } \Omega \setminus \Omega_S \end{cases}$$

and the pressure is linearly increased during the first 305 seconds until it reaches 600 MPa. Thus, the pressure generated by the equipment satisfies $P(0) = 0$ and

$$\frac{dP}{dt} = \begin{cases} \frac{120}{61} 10^6 \text{ Pas}^{-1}, & 0 < t \leq 305, \\ 0 \text{ Pas}^{-1}, & t > 305. \end{cases}$$

- ii) Process P2: The initial temperature is $T_0 = 40^\circ\text{C}$ in the whole domain Ω and the pressure is linearly increased (with the same slope as before) during the first 183 seconds until it reaches 360 MPa. Thus, the pressure generated by the equipment satisfies $P(0) = 0$ and

$$\frac{dP}{dt} = \begin{cases} \frac{120}{61} 10^6 \text{ Pa s}^{-1}, & 0 < t \leq 183, \\ 0 \text{ Pa s}^{-1}, & t > 183. \end{cases}$$

For each one of these four cases (solid/liquid, P1/P2) we compute the solution of the full model described in Section 5.1.3, we carry out the sensitivity analysis explained in Section 5.1.3 and we calculate the solutions for the simplified models described in Section 5.1.3.

Full model analysis Figure 5.2 shows the temperature distribution for both solid and liquid type food, under processes P1 and P2, at time $t = 15$ min. Time-averaged temperature distribution (i.e., the function $x \mapsto \mathcal{M}(T(x, \cdot); (0, t_f))$, for $x \in \Omega$) for the four cases is represented in Figure 5.3. The evolution of the sample mean temperature (i.e., the function $t \mapsto \mathcal{M}(T(\cdot, t); \Omega_F)$, for $t \in [0, t_f]$) is depicted in Figure 5.4, which also shows the evolution of the temperature at two points (see Figure 5.1): the first one, \mathbf{B}_1 , is located at the center of the sample (in the symmetry axis) and the second one, \mathbf{B}_2 , on the surface of the sample, at the same height as \mathbf{B}_1 .

These figures illustrate how the model captures the non-homogeneous temperature distribution inside the domain and the different behaviour between the solid and liquid cases. For instance, in the liquid case, the temperature distribution is more homogeneous than in the solid case, due to mass transfer. Therefore, the model and the numerical approximation of its solution is consistent with what is physically expected.

As already remarked in Ref. [219] for solid type foods, our numerical experiments show that for liquid type food it can also be interesting to use an initial food temperature lower than T_r (as done in process P1), in order to anticipate the temperature increase that results from compression. This allows us to get a more uniform process, avoiding big temperature gradients inside the food as well as much higher than T_r temperatures (we remind that one of the goals of the high-pressure technology is to process the food without using high temperatures, which degrade some of the main qualities of food). Figure 5.4 shows this behaviour for process P1.

Model sensitivity analysis According to Section 5.1.3, in order to evaluate the sensitivity of the models, we have generated $N = 10$ perturbed versions of the full models (with random perturbations of $\pm 5\%$). Table 5.1 summarizes the obtained results. The average AET of the mean temperature errors in the whole domain, defined in (5.1.13), is less than 1.15°C , which represents a relative error (see (5.1.14)) of 2.83%. Furthermore, the average AET_F of the mean temperature errors in the sample is less than 1.45°C and the relative error is less than 3.34%. We remark that both relative errors are of the order of the parameter perturbations, which shows the robustness of the models.

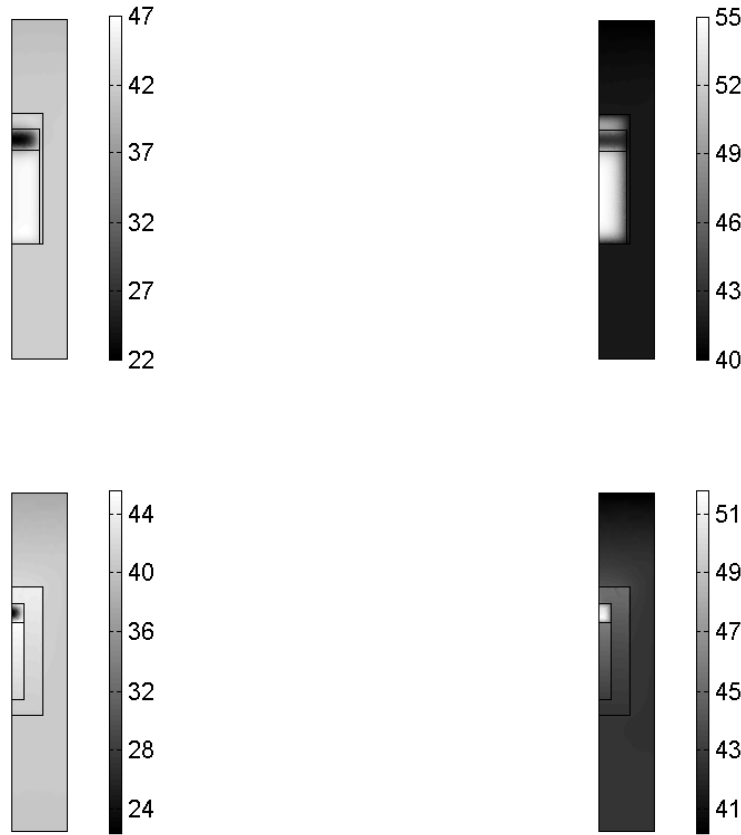


Figure 5.2: Temperature distribution ($^{\circ}\text{C}$) in the whole domain at $t = 15$ min for solid (**Top**) and liquid (**Bottom**) food cases after processes P1 (**Left**) and P2 (**Right**).

Simplified model analysis In this section we check the efficiency of the simplified models introduced in Section 5.1.3 and present the results in Table 5.2.

We can see that the mean temperature error in the sample is bigger than in the whole domain. This is due to the fact that the thermophysical parameters for the steel and the rubber cap are the same for the full and the simplified models. Therefore the differences are concentrated in the food and the pressurizing fluid. Nevertheless, the mean temperature relative errors in the food sample are quite acceptable, since the sensitivity analysis performed in Section 5.1.3 reveal an average mean relative error of the same order.

Comparing the processes, we observe that the error made in the three simplified models (SCC, LCC and LB) is less important in process P2 than in process P1. This is due to the fact that the range of pressure and temperature in P2 is smaller than in P1 and therefore, the approximation of parameters by their mean value is better in P2.

Comparing solid and liquid type foods, we observe that the errors made in the simplified models are smaller in the liquid cases than in the solid ones. The different size of the temperature ranges (larger for solid type food samples) that are obtained at each time explains this behaviour.

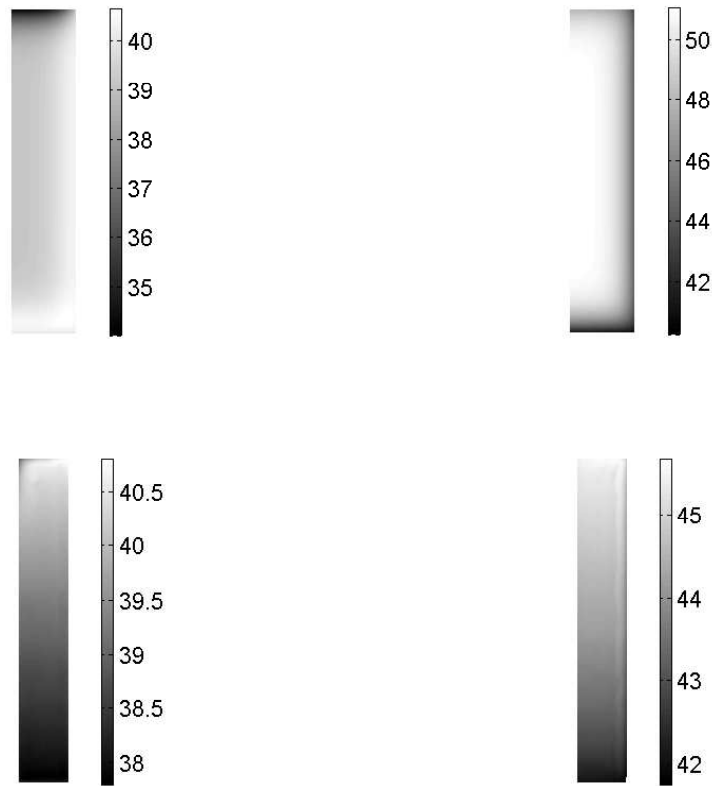


Figure 5.3: Time-averaged temperature distribution ($^{\circ}\text{C}$) during 15 min in the solid (**Top**) and liquid (**Bottom**) food sample after P1 (**Left**) and P2 (**Right**) processes.

Comparing the computational times we observe that in the solid case we have a reduction by a factor of 13 while in the liquid case this factor is less than 2. The reason is that the SCC model becomes linear but the LCC and LB remain nonlinear. In any case, LCC and LB have similar errors, but LB is faster and easier to implement on the computer.

The reduction of computational time combined with acceptable errors reveal that the simplified models are a good alternative to the full ones for optimization procedures.

5.1.4 Coupling of Enzymatic Inactivation and Heat–Mass Transfer Models

In this section we couple the heat transfer models presented in Section 5.1.3 with the kinetic equation (5.1.1). This allows us to get a model capable of describing non-homogeneous activity distribution inside the food. Moreover, we perform a numerical study of the impact of various HP–Thermal processes on the inactivation of three different enzymes: *Bacillus Subtilis* α –Amylase (BSAA), Lipoxxygenase (LOX) and Carrot Pectin Methyl–Esterase (CPE).

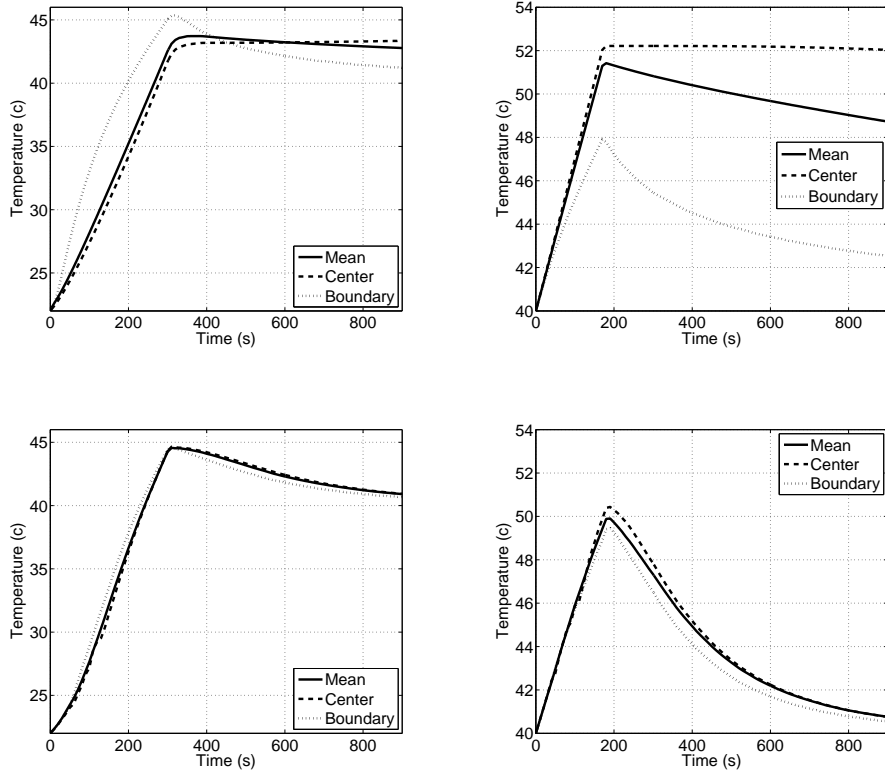


Figure 5.4: Evolution of the sample mean temperature (—), temperature in the center point \mathbf{B}_1 (- -) and in the boundary point \mathbf{B}_2 (...) (see Figure 5.1) during the processes P1 (**Left**) and P2 (**Right**) in the solid (**Top**) and liquid (**Bottom**) food samples.

Resulting activity equation

The activity distribution is expressed in a different way for solid type food (where the particles are considered to be motionless) and for liquid type food (where the particles are considered to move due to mass transfer).

Solid type foods For solid type foods we have supposed that the particles do not move. Thus, according to (5.1.4), the activity A of a particle located at the point $x \in \Omega_F$ at time t can be written as

$$A(x, t) = A(x, 0) \exp \left(- \int_0^t \kappa(P(\sigma), T(x, \sigma)) d\sigma \right). \quad (5.1.18)$$

The equipment pressure P (MPa) is a given function (we assume that the mass transfer pressure p is negligible compared to P) and the temperature T (K) is obtained by solving system (5.1.7).

Liquid type foods For liquid type foods, the particles move in the food domain Ω_F due to mass transfer. In this case, for each point $x \in \Omega_F$ we consider the trajectory of a

Table 5.1: MT: Mean temperature $\mathcal{M}(T; \Omega \times (0, t_f))$ ($^{\circ}\text{C}$); AET: Average mean temperature error ($^{\circ}\text{C}$) (see (5.1.13)); RAET: Relative average mean temperature error (%) (see (5.1.14)); MT_F , AET_F , RAET_F : Idem for the food sample, changing Ω by Ω_F .

Process	Food	Whole domain			Sample		
		MT	AET	RAET	MT_F	AET_F	RAET_F
P1	Solid	39.39	1.08	2.74	39.37	1.32	3.34
P2	Solid	41.46	1.14	2.75	49.25	1.45	2.93
P1	Liquid	39.77	1.07	2.68	39.06	1.05	2.70
P2	Liquid	40.49	1.15	2.83	43.95	1.17	2.67

Table 5.2: Results obtained for full and simplified models. MT: Mean temperature $\mathcal{M}(T; \Omega \times (0, t_f))$ ($^{\circ}\text{C}$); ET: Mean temperature error ($^{\circ}\text{C}$) (see (5.1.16)); RET: Relative mean temperature error (%) (see (5.1.17)); MT_F , ET_F , RET_F : Idem in the food sample; CT: Computational time (s) needed to solve the model.

Process	Model	Whole domain			Sample			CT
		MT	ET	RET	MT_F	ET_F	RET_F	
P1	SFull	39.39	—	—	39.37	—	—	53
P2	SFull	41.46	—	—	49.25	—	—	51
P1	SCC	39.71	0.30	0.77	41.38	1.88	4.77	4
P2	SCC	41.50	0.04	0.10	49.53	0.25	0.52	4
P1	LFull	39.77	—	—	39.06	—	—	3135
P2	LFull	40.49	—	—	43.95	—	—	4141
P1	LCC	39.90	0.16	0.41	39.95	0.81	2.07	2459
P2	LCC	40.48	0.03	0.06	43.94	0.09	0.20	2877
P1	LB	39.89	0.15	0.37	39.89	0.77	1.96	2196
P2	LB	40.47	0.03	0.08	43.94	0.10	0.22	2475

food particle that ends at point x . This trajectory X is the solution of

$$\begin{cases} \frac{dX}{dt}(t) = \mathbf{u}_F(X(t), t), & t \in (0, t_f), \\ X(t_f) = x, \end{cases} \quad (5.1.19)$$

where \mathbf{u}_F is the velocity field computed by solving system (5.1.10)–(5.1.11). Therefore, according to (5.1.4) again, the activity A of a particle located at the point $x \in \Omega_F$ at time t can be expressed as

$$A(x, t) = A(X(0), 0) \exp \left(- \int_0^t \kappa(P(\sigma), T(X(\sigma), \sigma)) d\sigma \right). \quad (5.1.20)$$

In this case, $T(X(t), t)$ is obtained by solving system (5.1.10)–(5.1.11). Here, $X(t)$ is the point that the trajectory X reaches at time t .

Discretization of the activity equation

Let us consider a time discretization given by $0 = t_0 < t_1 < \dots < t_n = t_f$ with step $\tau = t_i - t_{i-1} = \frac{t_f}{n}$ for $i = 1, 2, \dots, n$.

Solid type foods Given $x \in \Omega_F$ and denoting $\kappa_j(x) = \kappa(P(t_j), T(x, t_j))$, a numerical approximation of (5.1.18) can be obtained considering, for instance, the trapezoidal formula

$$A(x, t_n) \approx A(x, 0) \exp \left(-\frac{\tau}{2} \sum_{j=0}^{n-1} \left(\kappa_j(x) + \kappa_{j+1}(x) \right) \right). \quad (5.1.21)$$

The enzymatic activity is evaluated on a equally distributed mesh in the food sample domain Ω_F .

Liquid type foods For each point $x \in \Omega_F$ let X be the trajectory of a particle satisfying $X(t_f) = x$. We take an approximation $\{X_i\}_{i=0}^n$ of $\{X(t_i)\}_{i=0}^n$ defined by discretizing (5.1.19) with the following backward implicit scheme

$$\begin{cases} X_n = x \\ X_i = X_{i+1} - \tau \mathbf{u}_F(X_i, t_i), \quad i = n-1, n-2, \dots, 0. \end{cases} \quad (5.1.22)$$

The solution of (5.1.22) is computed by solving numerically the minimization problem

$$\min_{y \in \Omega_F} \|X_{i+1} - y - \tau \mathbf{u}_F(y, t_i)\|_2. \quad (5.1.23)$$

We use a Steepest-Descent algorithm starting from the solution of the backward explicit scheme

$$y_i^{(0)} = X_{i+1} - \tau \mathbf{u}_F(X_{i+1}, t_{i+1}),$$

that has to be iterated until reaching $y_i^{(m)} \in \Omega_F$ such that

$$\|X_{i+1} - y_i^{(m)} - \tau \mathbf{u}_F(y_i^{(m)}, t_i)\|_2 < 10^{-4} \|X_{i+1} - y_i^{(0)}\|_2.$$

By denoting $\kappa_j(X) = \kappa(P(t_j), T(X_j, t_j))$, a numerical approximation of (5.1.20) can be computed by considering

$$A(x, t_n) \approx A(X_0, 0) \exp \left(-\frac{\tau}{2} \sum_{j=0}^{n-1} \left(\kappa_j(X) + \kappa_{j+1}(X) \right) \right). \quad (5.1.24)$$

Again, the enzymatic activity is evaluated on a equally distributed mesh in the food sample domain Ω_F .

Enzymes considered for numerical simulation

For the numerical experiments, the following enzymes and corresponding inactivation rates have been considered:

Bacillus Subtilis α -Amylase (BSAA): It is an enzyme produced by a bacteria called *Bacillus Subtilis*. This bacteria, present in the ground, can contaminate food and in rare occasions cause intoxications. This enzyme catalyzes the hydrolysis of starch, generating sugars (as maltose) that can modify the taste of the food.

The inactivation rate κ is modelled using equation (5.1.2) with $P_{\text{ref}} = 500$ MPa, $T_{\text{ref}} = 313$ K, $\kappa_{\text{ref}} = 9.2 \times 10^{-2} \text{ min}^{-1}$, $B = 10097$ K and $C = -8.7 \times 10^{-4} \text{ MPa}^{-1}$. Interested readers can find more details about the experimental protocol and the parameters determination in Ref. [173].

Lipoxygenase (LOX): This enzyme is present in various plants and vegetables such as green beans and green peas. It is responsible for the appearance of undesirable aromas in those products.

Equation (5.1.3) is used to describe κ with $P_{\text{ref}} = 500$ MPa, $T_{\text{ref}} = 298$ K, $\kappa_{\text{ref}} = 1.34 \times 10^{-2} \text{ min}^{-1}$, $\Delta V_{\text{ref}} = -308.14 \text{ cm}^3 \text{ mol}^{-1}$, $\Delta S_{\text{ref}} = 90.63 \text{ J mol}^{-1} \text{ K}^{-1}$, $\Delta C_p = 2466.71 \text{ J mol}^{-1} \text{ K}^{-1}$, $\Delta \zeta = 2.22 \text{ cm}^3 \text{ mol}^{-1} \text{ K}^{-1}$, $\Delta \nu = -0.54 \text{ cm}^6 \text{ J}^{-1} \text{ mol}^{-1}$ (see Ref. [129] for more details).

Carrot Pectin Methyl-Esterase (CPE): Pectin Methyl-Esterase is an enzyme that is common to most vegetables. It can be present in vegetable juices producing low-methoxyl pectin. This process reduces juice viscosity and generates cloud loss (affecting juice flavor, color, texture and aroma). Here we concentrate on the Pectin Methyl-Esterase present in carrot juice (Carrot Pectin Methyl-Esterase).

In this case, we apply equation (5.1.3) to model κ with $P_{\text{ref}} = 700$ MPa, $T_{\text{ref}} = 323.15$ K, $\kappa_{\text{ref}} = 7.05 \times 10^{-2} \text{ min}^{-1}$, $\Delta V_{\text{ref}} = -44.0124 \text{ cm}^3 \text{ mol}^{-1}$, $\Delta S_{\text{ref}} = 168.4 \text{ J mol}^{-1} \text{ K}^{-1}$, $\Delta C_p = 1376.6 \text{ J mol}^{-1} \text{ K}^{-1}$, $\Delta \zeta = -0.0339 \text{ cm}^6 \text{ J}^{-1} \text{ mol}^{-1}$, $\Delta \nu = -0.1195 \text{ cm}^6 \text{ J}^{-1} \text{ mol}^{-1}$ (see Ref. [176]).

Numerical results

For the numerical tests, we consider $t_f = 15$ min and $n = 900$. In the solid case, the activity is evaluated over a 10000 point equally distributed mesh in Ω_F (we point out that the models for the solid case need a very low computational time and therefore we can use a very fine mesh). On the other hand, a 25 node equally distributed mesh is chosen for the liquid type food case (much more expensive than the solid case, from a computational point of view).

We present the enzymatic activity computed for the solid and liquid type food samples under processes P1 and P2. We are also interested in studying the sensitivity of the final enzymatic activity regarding the thermophysical parameters of the models (5.1.7) and (5.1.10)–(5.1.11). Finally, we analyze this activity when considering the simplified models presented in Section 5.1.3, instead of the full ones.

Full model analysis First of all, we focus on the enzymatic activity computed for the full models. As we can observe in Table 5.3 and Figure 5.5, the efficiency of processes P1

and P2 depends on the considered enzyme:

Table 5.3: Final activity for the full models and corresponding sensitivity analysis, for BSAA, LOX and CPE enzymes. MA: Mean activity in the full models at final time $t_f = 15$ min $\mathcal{M}(A(\cdot, t_f); \Omega_F)$ (%); AEA: Average mean final activity error ($^{\circ}\text{C}$) (see (5.1.25)).

Process	Food	BSAA		LOX		CPE	
		MA	AEA	MA	AEA	MA	AEA
P1	Solid	49.72	4.60	33.54	6.81	87.55	2.28
P2	Solid	26.43	5.01	67.18	6.43	93.86	0.52
P1	Liquid	53.57	4.02	36.95	7.45	88.17	2.40
P2	Liquid	48.18	3.97	83.82	2.51	95.16	0.28

- For enzyme BSAA we observe that process P2 is more efficient, since the final activity is smaller. However the difference between P1 and P2 in the liquid sample case (5%) is less important than in the solid one (23%). This is due to the fact that the difference of the mean temperatures between P1 and P2 is bigger in the solid type food sample (10°C) than in the liquid one (5°C) (see Table 5.1). This is consistent with the fact that BSAA is an enzyme sensible to high temperatures[173].
- For enzyme LOX, we remark that process P1 (600 MPa) is clearly more efficient than P2 (360 MPa) in both the liquid and solid samples, due to the sensitivity of this enzyme to high pressure[129].
- For enzyme CPE, processes P1 and P2 are not as efficient as before. In any case, better results are obtained with P1. This enzyme seems to be quite resistant to both processes.

Taking into account these results, we can not privilege process P1 or P2 as a general food treatment. In fact, for each kind of enzyme, a specific optimal process could be considered.

Figure 5.6 shows the final activity of enzyme LOX in the food sample for the solid and liquid cases and processes P1 and P2 (for the other enzymes, similar results are obtained). For the three considered enzymes, we can observe the following:

- In the solid case we obtain non-homogeneous activity distributions. Furthermore, for the considered enzymes and temperature-pressure ranges, we point out that the time-averaged temperature (see Figure 5.3) and final enzymatic activity distribution are related in the following way: the warmer (in time average) a zone is, the lower the final activity is. This behaviour can not be true for other enzymes and/or temperature-pressure ranges.
- In the liquid case, due to the mass transfer phenomenon, the enzymatic activity is more homogeneous than in the solid case.

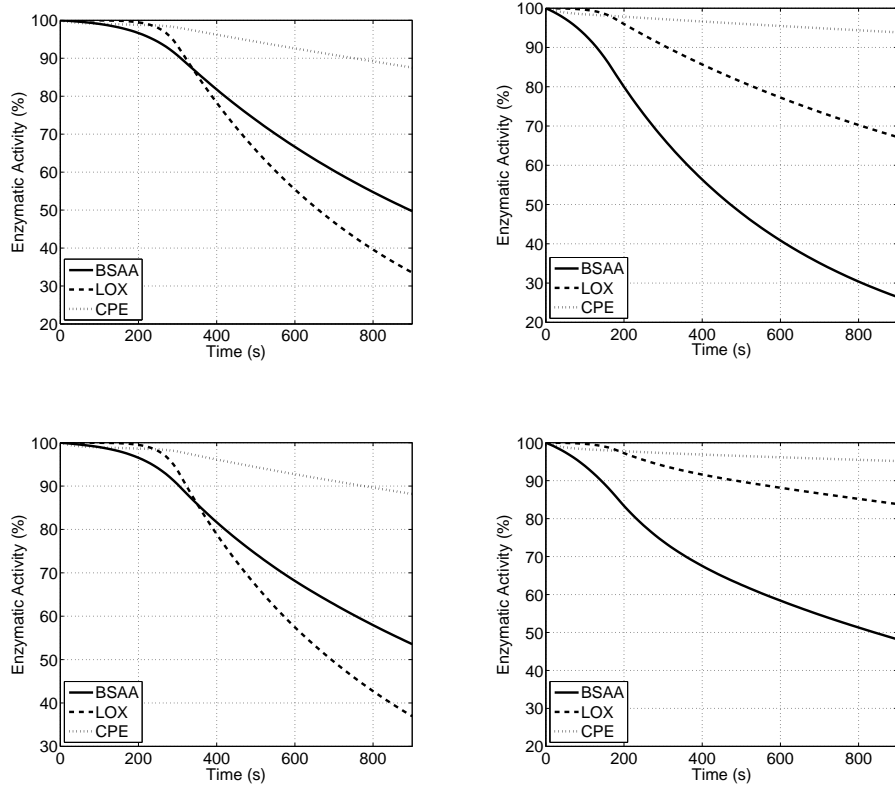


Figure 5.5: Enzymatic activity evolution (%) during processes P1 (**Leftt**) and P2 (**Rightt**) considering the solid (**Top**) and liquid (**Bottom**) food sample.

Sensitivity analysis We present results about final activity errors when considering the perturbed models generated in Section 5.1.3. Denoting by A_{per} the activity for the perturbed models, we define the average of the mean final activity error as

$$\text{AEA} = \frac{1}{N} \sum_{\text{per}=1}^N \mathcal{M}(|A(\cdot, t_f) - A_{\text{per}}(\cdot, t_f)|; \Omega_F). \quad (5.1.25)$$

The results obtained are reported in Table 5.3. We observe that perturbations of 5% of the model parameters (implying, as explained in Section 5.1.3, errors of 5% in the temperature of the food sample) generate final activity errors up to 7.45%. We deduce that the activity is relatively sensitive to pressure and temperature changes: the more sensitive to pressure and/or temperature the enzyme is, the more accurate the values of the thermophysical parameters and pressure curves should be.

Simplified model analysis In this section, we study the activity A_{sim} of the simplified models introduced in Section 5.1.3. We define the mean final activity error as

$$\text{EA} = \mathcal{M}(|A(\cdot, t_f) - A_{\text{sim}}(\cdot, t_f)|; \Omega_F). \quad (5.1.26)$$

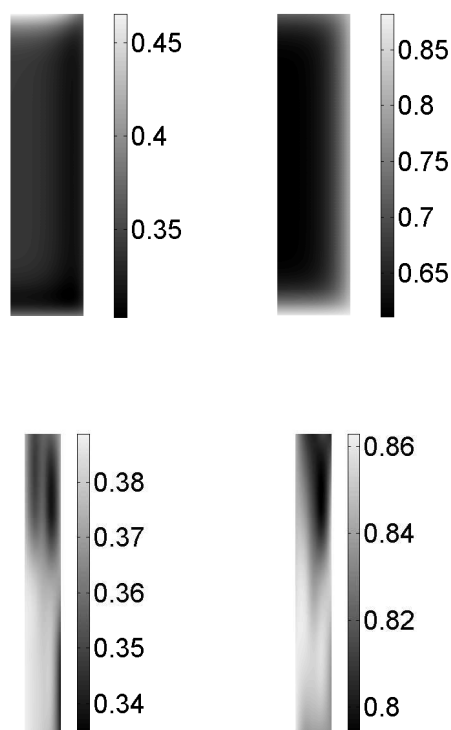


Figure 5.6: LOX activity distribution at $t = 15$ min in the solid (**Top**) and liquid (**Bottom**) food sample after P1 (**Left**) and P2 (**Right**) processes.

As we can observe in Table 5.4, the errors for process P2 are smaller than for process P1. Furthermore, simplified models LCC and LB are more accurate than SCC model. This is consistent with the differences in temperature distribution obtained between those models (see Table 5.2). Despite these errors, simplified models are suitable when computing the enzymatic activity, since they provide an acceptable information about it with a low computational cost.

5.1.5 Conclusions and Perspectives

The mathematical models described in this Section provide a useful tool to design and optimize processes based on the combination of thermal and high pressure processes in Food Technology. They take into account the heat and mass transfer phenomena and the enzymatic inactivation occurring during the process.

A sensitivity analysis has been developed in order to show the dependence of the solution regarding the thermophysical parameters, showing the robustness of the models.

Several simplified versions of the full models have been also proposed. When comparing them to the corresponding full model the results are quite similar. Therefore, since the simplified models need less computational time to be solved, they can be suitable for optimization procedures (which usually need to compute the solution of the corresponding

Table 5.4: Final activity for the full and simplified models for BSAA, LOX and CPE enzymes. MA: Mean activity at final time $t_f = 15$ min $\mathcal{M}(A(\cdot, t_f); \Omega_F)$ (%); EA: Mean activity error ($^{\circ}\text{C}$) (see (5.1.26)).

Process	Model	BSAA		LOX		CPE	
		MA	EA	MA	EA	MA	EA
P1	SFull	49.72	—	33.54	—	87.55	—
P2	SFull	26.43	—	67.18	—	93.86	—
P1	SCC	42.29	7.44	28.35	5.20	86.22	1.33
P2	SCC	25.47	0.96	66.07	1.11	93.76	0.10
P1	LFull	53.57	—	36.95	—	88.17	—
P2	LFull	48.18	—	83.82	—	95.16	—
P1	LCC	50.76	2.81	35.20	1.75	87.77	0.40
P2	LCC	47.77	1.14	83.55	0.65	95.14	0.06
P1	LB	50.56	3.04	34.97	2.00	87.72	0.45
P2	LB	47.75	2.23	83.55	1.31	95.13	0.12

model many times for different data).

All these numerical results show that there is not a general optimal treatment. For each particular kind of food and high pressure equipment we propose to carry out the following steps:

- i) Identify the most important enzymes to be inactivated and get, for each one of them, a kinetic equation describing the evolution of their activity in terms of pressure and temperature (see Sections 5.1.2 and 5.1.4).
- ii) Choose a suitable model describing the distribution of temperature in the food and solve it numerically (see Section 5.1.3).
- iii) Use this distribution of temperature as input data for the selected kinetic equations of the enzymes, in order to get their final activities after the thermal-HP process (see Section 5.1.4).
- iv) Perform optimization techniques (which may need to carry out several numerical experiments changing initial temperature, applied pressure, etc.) in order to reduce the enzymatic activities without using high temperatures (whose drawbacks are described in Section 5.1.3). This task may be performed with the help of optimization software such as *Global Optimization Platform* presented in Chapter 5.2.4.

5.2 Estimation of the dynamics of protein folding processes from free energy computations

5.2.1 Introduction

One of the main causes of organoleptic degradation of food is the enzyme activities. Reducing this activity can help increase the shelf life of food. For this purpose, the understanding of the inactivation of enzymes is a key point. Taking into account that enzymes are proteins which are active when their structures are in a so-called folded state and inactive when they are in an unfolded state, the simulation of protein configurations and kinetics is a useful tool for the food industry [130].

Proteins are molecules composed by a sequence of amino acids that are involved in many biological processes such as the catalysis of chemical reactions, being part of a cell's membrane, acting as signaling devices, etc. [167]. As they are synthesized inside the cells, they start folding in space, *i.e.*, the protein's shape goes from an unfolded state (the protein is inactive) to a folded state (the protein can perform its task). Techniques such as X-ray crystallography or nuclear magnetic resonance [167] allow us to observe the three dimensional structure of proteins. However, due to technical limitations, it is not easy to see the variations in the protein structure occurring during the folding process inside a living organism. Thus, we are interested in practical and reliable methods to simulate those changes.

We generally consider mathematical models based, for instance, on Classical and Statistical Mechanics [203, 204, 259]. The coefficients of those models are calibrated with results from laboratory experiments. From an experimental point of view, protein folding can be started, for instance, after applying changes in temperature or pressure [130] to a protein under study. Those changes disturb the protein's conformational equilibrium, necessary to its biological function.

In this Section, we present a mathematical model based on a partial differential equation (PDE) to represent the folding process of the protein BBL¹ after it is subjected to a sudden change in its temperature. Numerical solutions of this model allow us to recover a laboratory-verifiable property called *relaxation rate*. The considered PDE is a diffusion equation with a diffusion coefficient dependent on temperature and a reaction coordinate (which can be interpreted as a spatial component). We find the diffusion coefficient by means of a global optimization approach using data coming from experiments [226, 58].

In Section 5.2.2, we recall some of the basics of molecular modeling. Section 5.2.3 introduces the considered PDE model and its parameters. Then, in Section 5.2.4, we present two optimization problems used to determine the particular PDE's diffusion coefficient. Finally, in Section 5.2.5, we give the optimized results for two possible functional forms of this diffusion coefficient.

¹<http://www.pdb.org/pdb/explore.do?structureId=1BBL>

5.2.2 Some preliminaries on molecular modeling

In this section we introduce the relevant terminology and algorithms that serve to set up the method developed in Section 5.2.3. In particular, we discuss how incomplete but computationally feasible Monte Carlo strategies together with coarse-grained models and the solution of a PDE, lead to a practical way to recover relaxation rates (a concept that will be expanded upon in Section 5.2.3). In addition to that, we introduce the concepts of reaction coordinate and free energy.

Monte Carlo approach

The state of a protein when it is modeled as a classical mechanical system is determined by the phase-space point (\mathbf{x}, \mathbf{p}) , with \mathbf{x} belonging to the **conformational space** \mathcal{D} and \mathbf{p} being the corresponding momentum. The forces of interaction between particles are determined by a real-valued function $U(\mathbf{x})$ called the **potential energy function**.

The most straight forward way to compute the relaxation rates would consist in integrating the equations of motion of the system, an approach known as Molecular Dynamics (MD) [165]. However, MD simulations of proteins are outside of the scope of the capabilities of current technologies [39] due to the time scales at which proteins fold. Hence the need to resort to alternative techniques such as Monte Carlo (MC) strategies [172], that we will now briefly explain.

The MC strategy used in this work explores the graph of U corresponding to a protein by performing a self-avoiding random walk in \mathcal{D} . This random walk is biased by U so that high probability (or, equivalently, low energy) conformations are more likely to be visited. In practice, this means that the simulated protein goes from an unfolded state to its stable folded state at an adequate temperature.

A drawback of the MC method is that information regarding the velocities of each particle is not computed and, consequently, all kinetic properties including relaxation rates remain unknown. Nevertheless, the MC approach is computationally feasible and provides us with valuable information that we can complete by further extending our assumptions and postulating that the time evolution of the folding process can be described by diffusion along a reaction coordinate. This method also allows us to consider simplified potential energy functions so that we can rely on coarse-grained models that are computationally less expensive to simulate.

Coarse-grained model

We consider a model of a protein in which each amino acid is represented by a **coarse-grained (CG) site**. A protein in this model is regarded as a chain of N such CG sites.

Definition 5 *The **native conformation** of a protein is the point*

$$\mathbf{x}^{nat} = (\mathbf{x}_1^{nat}, \dots, \mathbf{x}_N^{nat}) \in \mathcal{D},$$

where $\mathcal{D} = \mathbb{R}^{3N}$ and² $\mathbf{x}_k^{\text{nat}} \in \mathbb{R}^3$ denotes the position of the k -th CG site³.

The distance between two arbitrary CG sites i and j in the native conformation is denoted by

$$d_{ij}^{\text{nat}} = |\mathbf{x}_i^{\text{nat}} - \mathbf{x}_j^{\text{nat}}|. \quad (5.2.1)$$

Potential energy function

The potential energy function considered during this work belongs to a class of functions called **structure-based (or native-centered) potentials** [101]. Let P be a protein and let \mathbf{x}^{nat} be its native conformation. The **actual spatial conformation** of P at an arbitrary instant is denoted by

$$\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_N) \in \mathcal{D},$$

where $\mathbf{x}_k \in \mathbb{R}^3$ stands for the position of the k -th CG site of P . The potential energy function is defined by

$$U(\mathbf{x}) = \sum_{\substack{i=1 \\ j>i}}^N u_{ij}(\mathbf{x}), \quad (5.2.2)$$

where the **pairwise potential energy**

$$u_{ij}(\mathbf{x}) = u_{ij}(\mathbf{x}; \mathbf{x}^{\text{nat}}, d_{\max}, \ell)$$

is a real-valued function of \mathbf{x} that depends on \mathbf{x}^{nat} and on two distances d_{\max} and ℓ (measured in Ångstroms) typically taking values in the intervals $(3.8, 10]$ and $(0, 1]$, respectively. The functional form of the pairwise potential is

$$u_{ij}(\mathbf{x}) = -1 + \left(\frac{g_{ij}(\mathbf{x})}{\ell} \right)^2, \quad (5.2.3)$$

where

$$g_{ij}(\mathbf{x}) = \begin{cases} 0 & \text{if } j = i + 1, \\ d_{ij} - d_{ij}^{\text{nat}} & \text{if } |d_{ij} - d_{ij}^{\text{nat}}| < \ell \text{ and } j = i + 2, \\ d_{ij} - d_{ij}^{\text{nat}} & \text{if } |d_{ij} - d_{ij}^{\text{nat}}| < \ell, \quad j = i + 3, \\ & \text{and } \text{sgn}(\det(V)) = \text{sgn}(\det(V^{\text{nat}})), \\ d_{ij} - d_{ij}^{\text{nat}} & \text{if } |d_{ij} - d_{ij}^{\text{nat}}| < \ell, \quad j \geq i + 4, \quad \text{and } d_{ij}^{\text{nat}} \leq d_{\max}, \\ \ell & \text{otherwise,} \end{cases} \quad (5.2.4)$$

with $d_{ij} = |\mathbf{x}_i - \mathbf{x}_j|$ and d_{ij}^{nat} as defined in (5.2.1). A graphical representation of the pairwise potential $u_{i,i+2}$ is displayed in Figure 5.7.

²The dimension of \mathcal{D} is actually less than $3N$. We can parameterize the conformational space using so-called internal coordinates [165] using the least number of degrees of freedom. However, the redundant parameterization $\mathcal{D} = \mathbb{R}^{3N}$ leads to simpler formulas in our case.

³This information can be found, for instance, in the Protein Data Bank <http://www.pdb.org>.

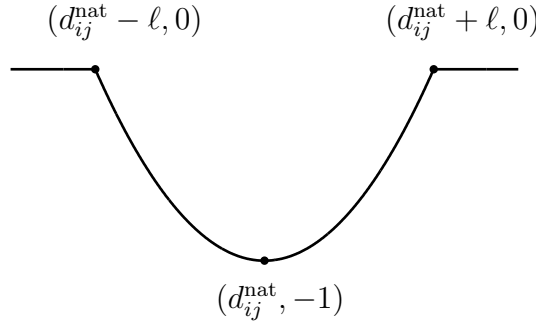


Figure 5.7: Graph of the pairwise potential $u_{i,j}$ as a function of $d_{i,j}$ in the case $j = i + 2$.

Remark 6 The condition $\text{sgn}(\det(V)) = \text{sgn}(\det(V^{nat}))$ in (5.2.4) is known in the literature as a **chiral constraint** [165]. Its purpose is to discriminate between a molecule and its mirror image.

Remark 7 The potential energy function U takes values in the interval $[a, b] \subset \mathbb{R}$, where

$$a = U(\mathbf{x}^{nat}) \leq U(\mathbf{x}) \leq 0 = b \quad \text{for all } \mathbf{x} \in \mathcal{D}.$$

Reaction coordinates and free energy

Definition 8 A **reaction coordinate** is a coordinate system $\xi : \mathcal{D} \rightarrow [a, b]$ that describes the progress of the protein folding process.

Definition 9 For systems with constant number of particles, volume and temperature (T), we can define the **free energy** as

$$F(x) = U(x) - T S(x), \quad (5.2.5)$$

where x is the value of the reaction coordinate, U is the potential energy, and S is the **configurational entropy** that is related to the density of states for each value of the reaction coordinate [47].

Remark 10 The free energy balances U against the entropic term TS . For a fixed temperature, both of these terms decrease as the native state is reached.

The choice of reaction coordinate is not unique and, in this work, we take ξ equal to the potential energy function U . This essentially measures how close the protein is to the native state. For this choice, the expression of F in (5.2.5) becomes

$$F(U) = U - T S(U). \quad (5.2.6)$$

Note that internal energies U and their corresponding configurational entropies $S(U)$ are obtained via MC molecular simulations, as described in Section 5.2.2. The free energy

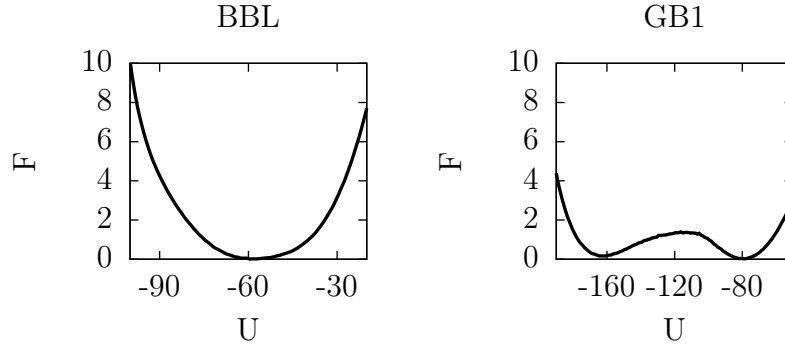


Figure 5.8: Free energy versus potential energy for the proteins BBL (left) and GB1 (right) at their transition temperatures (328 K and 358 K, respectively).

profile for BBL at the temperature where the transition from unfolded to the folded state happens is depicted in Figure 5.8 (left).

Most proteins present peaks in their free energy profiles between the unfolded states and the folded state. These peaks are referred to as **free energy barriers**. An example of a free energy barrier can be observed in Figure 5.8 (right) around $U = -120$.

Relaxation rate

Let T_1 and T_2 be two different temperatures. We define the **relaxation rate** k of a protein corresponding to the pair of temperatures (T_1, T_2) as

$$k = \frac{\log(2)}{t^*}, \quad (5.2.7)$$

where t^* is the instant where the protein is half-way to equilibrium at temperature T_1 after its temperature has been changed to T_2 .

This definition is imposed by the criteria used to make the experimental measurements that were available to us.

Note that the general definition given in the literature (see, for instance, [39]) coincides for two-state proteins (which is not the case for BBL) with the one given above, when T_1 is a temperature where the protein is folded and T_2 is a temperature where it is unfolded. Since our results involve matching simulations with experimental data, they are not affected by this discrepancy.

5.2.3 PDE model

On one hand, for a given protein and several temperatures T_1, \dots, T_L , we have the corresponding relaxation rates $k_1^{\text{exp}}, \dots, k_L^{\text{exp}} \in \mathbb{R}$, expressed in s^{-1} , coming from experiments, as described in Section 5.2.3. In the sequel, we denote $k^{\text{exp}} = (k_1^{\text{exp}}, \dots, k_L^{\text{exp}}) \in \mathbb{R}^L$. On the other hand, we have free energy profiles for T_i with $i \in \{1, \dots, L\}$ coming from MC simulations.

As explained in Section 5.2.2, MC simulations can only give conformational and energetic data. In principle, we cannot use these results to obtain any relaxation rates that

can be compared with k^{exp} . However, assuming that the kinetics of the protein folding process can be described by a diffusion process along a reaction coordinate, we can compute relaxation rates k^{sim} and compare them with k^{exp} . The point at issue is to find a suitable diffusivity profile that fits the folding kinetics. In the following sections, we detail these computations.

Temperature-jump experiments

We assume, without loss of generality, that $T_1 < \dots < T_L$. The procedure to obtain the experimental relaxation rates k^{exp} is as follows:

1. For each T_i , we pick a neighboring temperature T_j , with $j = i + 3$ if $i < L - 3$ and $j = i - 3$ elsewhere.
2. Next, the system is perturbed until it reaches T_j .
3. Then, we observe how long it takes the system to get half-way to equilibrium at temperature T_i .
4. Finally, we obtain the relaxation rate k_i^{exp} by using (5.2.7).

This procedure is known as a *temperature jump* (or *T-jump*) experiment. In order to simulate it, we solve the Smoluchowski equation to obtain the evolution of the states of the system after its temperature has been suddenly changed. In the following section, we explain precisely how this is achieved.

The Smoluchowski equation

Performing the *T-jump* procedure requires that we know at each instant the probability density function of the states of the system when it is carried away from equilibrium. In order to mathematically model and simulate these T-jump experiments, our assumption (following [250]) is that the time evolution of the system is described by a diffusion process along the reaction coordinate⁴ x in the free energy given by (5.2.6). Under this assumption, the time-dependent probability density function u can be obtained by solving the Smoluchowski equation.

The Smoluchowski equation is a parabolic partial differential equation that arises as a special case of the *Fokker-Planck* equation [237]. It is given by

$$u_t = J_x, \quad x \in (a, b), t > 0, \quad (5.2.8)$$

$$J(x, t) = 0, \quad x \in \{a, b\}, t > 0, \quad (5.2.9)$$

$$u(x, 0) = u_0(x), \quad x \in (a, b), \quad (5.2.10)$$

where $u(x, t)$ is the probability density function of the distribution of states with reaction coordinate x at time t . The flux is

$$J(x, t) = D(x) p(x) \left(\frac{u(x, t)}{p(x)} \right)_x, \quad (5.2.11)$$

⁴Recall from Section 5.2.2 that we established $U : \mathcal{D} \rightarrow [a, b]$ as a natural reaction coordinate.

with $D(x)$ being the diffusivity profile and $p(x)$ the equilibrium probability (both are temperature-dependent), and $u_0(x)$ the initial probability density function. At temperature T_i , we assume (see [280]) that $p(x) = p_i(x)$ with

$$p_i(x) = \frac{e^{-F_i(x)}}{\int_a^b e^{-F_i(x)} dx}, \quad i \in \{1, \dots, L\},$$

where $F_i(x)$ is the free energy at temperature T_i for the state with reaction coordinate $x \in [a, b]$. We also set $u_0(x) = p_j(x)$ in (5.2.10).

It can be proven [237, Section 6.1] that, for an initial probability density function u_0 , a solution $u(x, t)$ of (5.2.8)-(5.2.10) satisfies

$$\lim_{t \rightarrow +\infty} u(x, t) = p(x). \quad (5.2.12)$$

Numerical solution

In order to simplify the numerical discretization process, we rewrite (5.2.8) as a convection-diffusion equation by setting

$$u(x, t) = v(x, t) p(x).$$

In that case, we obtain

$$v_t = Dv_{xx} + (D_x - DF_x)v_x, \quad (x, t) \in (a, b) \times (0, +\infty), \quad (5.2.13)$$

subject to the boundary conditions

$$v_x(a, t) = v_x(b, t) = 0, \quad t > 0, \quad (5.2.14)$$

$$v(x, 0) = \frac{u_0(x)}{p(x)}, \quad x \in (a, b). \quad (5.2.15)$$

In order to discretize this boundary value problem, we use an equally-spaced spatial and time mesh of step sizes h and k , respectively. Thus,

$$\begin{aligned} x_i &= a + ih \quad \text{and} \quad t_j = jk, \\ \text{for } i &\in \{0, 1, \dots, N\} \quad \text{and} \quad j \in \{0, 1, \dots\} \quad \text{with} \quad h = (b - a)/N. \end{aligned}$$

To solve (5.2.13)-(5.2.15), we use the following time-implicit convection-centered and diffusion-centered finite difference scheme

$$\frac{v_i^{j+1} - v_i^j}{k} = D_i \frac{v_{i+1}^{j+1} - 2v_i^{j+1} + v_{i-1}^{j+1}}{h^2} + \left(\frac{D_{i+1} - D_{i-1}}{2h} - D_i \frac{F_{i+1} - F_{i-1}}{2h} \right) \frac{v_{i+1}^{j+1} - v_{i-1}^{j+1}}{2h},$$

where $v_i^j = v(x_i, t_j)$, $D_i = D(x_i)$ and $p_i = p(x_i)$. Considering this discretization, the boundary conditions are:

$$v_{-1}^{j+1} = v_1^{j+1}, \quad v_{N-1}^{j+1} = v_{N+1}^{j+1}, \quad \text{and} \quad v_i^0 = \frac{u_0(x_i)}{p(x_i)}.$$

Since we are interested in obtaining the relaxation rates, we run our numerical scheme until the solution converges to $p(x)$. Or, equivalently by (5.2.12), until $v(x, t)$ converges to 1. We use a precision of $\varepsilon = 10^{-7}$ to establish convergence in $L^2(a, b)$.

Computation of simulated relaxation rate

The simulated relaxation rate k^{sim} for the considered temperatures is derived from the numerical solution of (5.2.8)-(5.2.10) described in the previous section.

More precisely, we first define the function

$$A(t) = \int_a^z u(x, t) dx,$$

where $z \in (a, b)$ is some temperature-independent threshold that determines whether the protein is folded ($x \in [a, z]$) or unfolded ($x \in (z, b]$). The choice of z comes from experimental measurements. In the case of BBL, this value was set to -59 . More detail can be found, for instance, in [164].

For each time t_j defined in Section 5.2.3, we approximate $A(t_j)$ using the trapezoidal integration rule and we denote the approximation by A_j . Let

$$A_\infty = \lim_{t \rightarrow \infty} A(t) = \int_a^z p_i(x) dx.$$

Then, we construct the piecewise linear interpolant $g : \mathcal{A} \rightarrow [0, +\infty)$ corresponding to the mapping $A_j \mapsto t_j$, where $\mathcal{A} = \{A(t) \in [0, 1] : t \in [0, +\infty)\}$. Finally, $t^* = g\left(\frac{A_\infty + A_0}{2}\right)$ and the relaxation rate k^{sim} follows from (5.2.7).

Accuracy, stability, and execution time

In order to check the accuracy and stability of our numerical scheme presented in Section 5.2.3, we study the evolution of a Gaussian distribution under (5.2.8)-(5.2.10) as a benchmark problem.

We consider $a = -1$, $b = 1$, constant diffusivity $D = 10^3$, $z = 0$, and the probability density functions

$$p(x) = \frac{e^{-10(x+0.3)^2}}{\int_{-1}^1 e^{-10(x+0.3)^2} dx}, \quad u_0(x) = \frac{e^{-10x^2}}{\int_{-1}^1 e^{-10x^2} dx}.$$

Considering meshes with different space and time step sizes $h \in \{10^i\}_{i=-5, \dots, 0}$ and $k \in \{10^i\}_{i=-7, \dots, -2}$, we compute the relaxation rate denoted by $k^{\text{sim}}(h, k)$. In Table 5.5, we report the percentage of difference of $k^{\text{sim}}(h, k)$ with respect to the most refined case $k^{\text{sim}}(10^{-5}, 10^{-7})$ computed as

$$100 \left(1 - \frac{k^{\text{sim}}(h, k)}{k^{\text{sim}}(10^{-5}, 10^{-7})} \right).$$

Table 5.6 shows the elapsed execution times for the different values of h and k .

Those data helped us to evaluate the ratio between numerical reliability of the solution and execution speed. From those results, we decided to take $h = 10^{-2}$ and $k = 10^{-7}$ for the simulations described in the Section 5.2.3 and the optimizations to be explained subsequently.

Remark 11 *It is also important to note that the solutions obtained for different temperatures are independent of each other. Consequently, it is possible to solve (5.2.8)-(5.2.10) in parallel for all the temperatures and thus reduce the overall execution speed [4].*

$\log_{10} h \backslash \log_{10} k$	-7	-6	-5	-4	-3	-2
-5	0	1.29	12.16	62.78	95.05	99.49
-4	0.01	1.30	12.17	62.78	95.05	99.49
-3	0.10	1.39	12.23	62.79	95.05	99.49
-2	0.96	2.24	12.82	62.87	95.05	99.49
-1	6.05	7.21	16.96	63.41	95.06	99.49
0	93.58	95.27	95.29	95.27	96.56	99.51

Table 5.5: Percentage of difference between $k^{\text{sim}}(h, k)$ and $k^{\text{sim}}(10^{-5}, 10^{-7})$ obtained during the experiments described in Section 5.2.3

$\log_{10} h \backslash \log_{10} k$	-7	-6	-5	-4	-3	-2
-5	183.80	21.93	5.51	5.57	5.50	5.48
-4	16.65	1.64	0.44	0.40	0.48	0.48
-3	1.58	0.18	0.05	0.05	0.05	0.05
-2	0.44	0.05	0.01	0.01	0.01	0.01
-1	0.31	0.04	0.01	0.01	0.01	0.01
0	0.81	0.24	0.03	0.01	0.01	0.01

Table 5.6: Mean execution time (s) obtained during the experiments described in Section 5.2.3

5.2.4 Estimation of the diffusivity profile

As mentioned in Section 5.2.3, we want to find the diffusivity profiles of our model that better fits laboratory results. Our goal is to find the optimal diffusivity profiles for the BBL protein and, to do so, we used experimental relaxation rates obtained at 38 different temperatures ranging from 280K to 333K.

The diffusivity profile D in (5.2.8) was assumed to belong to two kinds of functions. Each one depending on a set of parameters which, once chosen, determines, in a univocal way, an expression for D into the Smoluchowski equation.

we used the MSA methodology presented in Chapter 2 combined with a Pattern Search method for the core optimization algorithm [155]. Our fitness function is defined by

$$J_{k^{\text{exp}}}(D) = \sum_{i=1}^L \left(\log_{10} \left(\frac{k_i^{\text{exp}}}{k_i^{\text{sim}}(D)} \right) \right)^2, \quad (5.2.16)$$

where $k_i^{\text{sim}}(D)$ comes from the numerical solution of (5.2.8) when considering D as the diffusivity profile, as explained in Section 5.2.3.

Exponential diffusivity profile

First, as done in other works such as [206], we consider profiles depending only on temperature and based on the Arrhenius expression

$$D_{k_0, E_a}(T) = k_0 \exp \left\{ -\frac{N E_a}{RT} \right\}, \quad (5.2.17)$$

where $k_0 > 0$ is a temperature-independent constant of the same units as D , E_a is the so-called *activation energy per amino acid* [J mol⁻¹], N is the *number of amino acids*, and R is the *gas constant*, approximately equal to 8.314472 J K⁻¹ mol⁻¹.

Note that the parameters $(k_0, E_a) \in M = [10^4, 10^{12}] \times [0.5, 3.5]$ completely determine the expression (5.2.17) for a given protein (which imposes the value of N) and temperature. Consequently, we want to find

$$\operatorname{argmin}_{(k, E) \in M} J_{k^{\exp}}(D_{k, E}).$$

5-point interpolated profile

A more accurate form of the diffusivity profile should be dependent on the temperature *and* the reaction coordinate. This is because the free energy function should play a role in the folding rates and, consequently, in the assumed diffusive dynamics of the protein folding process [250, 27]. Therefore, we postulate that the diffusivity profile is a continuous function

$$D : [a, b] \times [T_1, T_L] \rightarrow [10^3, 10^9].$$

This function is characterized by choosing five points in the graph of D such that four of them lie in the corners of the rectangle $[a, b] \times [T_1, T_L]$ and another one lies in the rectangle's interior. Thus, the diffusivity profile is determined by a set of 7 parameters denoted by $\{PI_i\}_{i=1}^7 \in \Gamma$ (a subset of \mathbb{R}^7 defined by the relevant restrictions for each parameter). The function is constructed by using the MATLAB command `TriScatteredInterp` command, which computes the Delaunay triangulation of the points in the domain of D and then builds the corresponding linear interpolants. The diffusivity profile associated to $\{PI_i\}_{i=1}^7 \in \Gamma$ is denoted by D_{PI_1, \dots, PI_7} . In that case, we want to find

$$\operatorname{argmin}_{(PI_1, \dots, PI_7) \in \Gamma} J_{k^{\exp}}(D_{PI_1, \dots, PI_7}).$$

5.2.5 Results

The total number of evaluations of the fitness function (5.2.16) used by the considered MSA for determining the two diffusivity profiles presented above is given in Table 5.7. The second column of this table corresponds to the time of the computations on an 8-core Intel Xeon E5620 computer at 2.40 GHz. The convergence of both optimization experiments is shown in Figure 5.9.

	Evaluations	Time (approx.)
Exponential profile	18400	5h 30m
5-point interpolated profile	14000	4h 10m

Table 5.7: Number of evaluations of the fitness function and total time for the two diffusivity profiles under consideration.

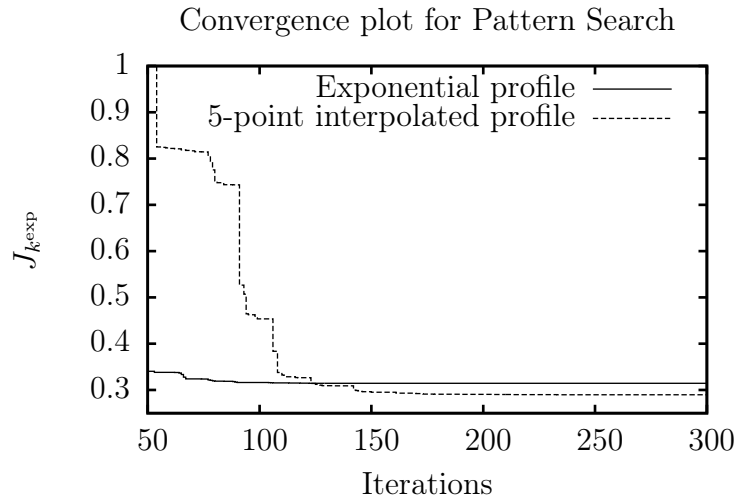


Figure 5.9: Convergence history of the exponential profile and the 5-point interpolated profile optimization processes: Fitness vs. Iteration function value J_k^{exp} .

The resulting optimal diffusivity profiles and their fitness values are shown in Figures 5.10 and 5.11. The best element found when considering the exponential profile has a fitness function value of 0.314587 whereas in the 5-point interpolated profile case the best value of $J_{k^{\text{exp}}}$ is 0.289033. Thus, the best 5-point interpolated profile is 8.12% more accurate than the best exponential profile in predicting relaxation rates. This was to be expected (in light of what was explained in Section 5.2.4) because the 5-point interpolated profile takes into account the reaction coordinate whereas the exponential profile does not.

5.2.6 Conclusions and Perspectives

During this work, we have presented two methods, based on the optimization of the diffusivity profile of the Smoluchowski equation, to determine some relaxation rates of the BBL protein. The understanding of the protein folding/ unfolding process is an important tool for the study of enzyme inactivation and, thus, for food safety. We have found that, for BBL, the optimized reaction coordinate-dependent diffusivity profiles provide more accurate results for the estimation of relaxation rates than the optimized diffusivity profiles that only take temperature into account.

Future extensions of our work could be:

- Use more data coming from laboratory experiments and/or small time-scale molecular dynamics simulations to obtain diffusivity profiles for a wider array of proteins, possibly having significantly different kinetics.
- Try multiobjective optimization techniques on the diffusivity profiles of proteins whose folding processes are known to share similar qualities.
- Consider reaction coordinates of the form $\xi : \mathcal{D} \rightarrow \mathbb{R}^s$ with $s \in \mathbb{N}$ small. This approach could be generalized in order to improve the ability of the model to capture the multidimensional nature of the folding process.
- Extend our *in silico* temperature-jump experiments to pressure-jump experiments. In that case, we could study how the diffusivity profiles (and, consequently, our approximation of the protein folding kinetics) depend on pressure in addition to temperature and reaction coordinate.
- Study possible physical interpretations of the interior point found in $(a, b) \times (T_1, T_L)$ as a part of the optimization process (see Section 5.2.4).
- Estimate the minimum number of experiments needed to characterize the diffusivity profiles (via (5.2.16)).

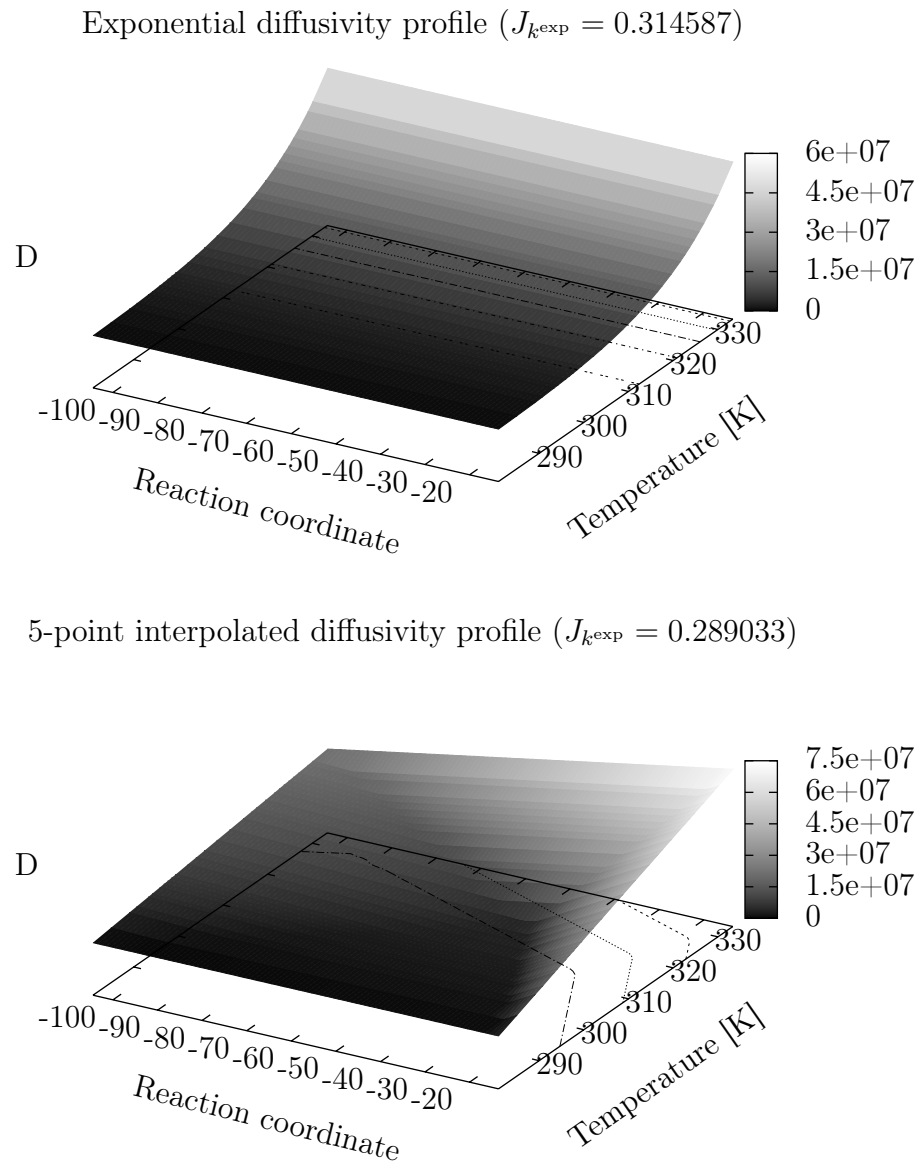


Figure 5.10: Diffusivity profiles obtained through **(TOP)** the exponential profile and **(BOTTOM)** the 5-point interpolated profile optimization processes.

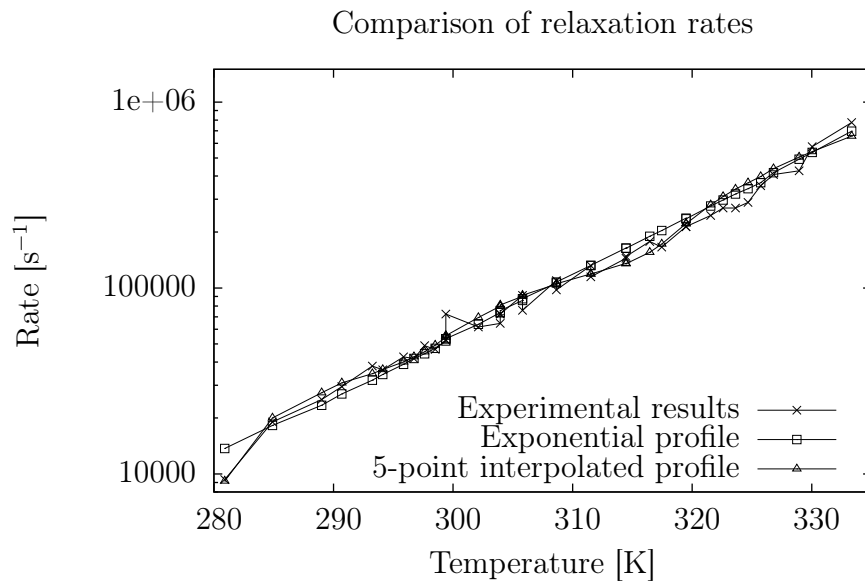


Figure 5.11: Results comparing k^{exp} with k^{sim} corresponding to the optimized exponential and 5-point interpolated diffusivity profiles.

Chapter 6

Modeling, Simulation and Optimization of a Polluted Water Pumping Process in the Open Sea

Abstract - *In this Chapter, we first discuss the modeling and simulation of the motion of oil spots in the open sea, and the effect on the pollutant concentration when a Skimmer ship (i.e. pumping ship) follows a pre-assigned trajectory to remove the pollutant. We assume here that the oil spots motion is due to the coupling of diffusion, the transport from the wind, sea currents and pumping process and the reaction due to the extraction of oil, implying that the mathematical model will be of advection-reaction-diffusion type. Our discussion includes the description of a parallelization of the selected numerical procedure. We present some results of numerical experiments showing that indeed the parallelization makes the model evaluation more efficient. The second objective of this Chapter is to introduce a method to find the optimal trajectory of the Skimmer ship, in order to pump the maximum quantity of pollutant on a fixed time period. The trajectory of the ship is directly modeled by considering a finite number of interpolation points for cubic splines. The optimization problem is solved by using the GMSA presented in Section 2.3.3. Finally, we check the efficiency of our approach by solving several numerical examples considering various shapes of oil spots based on real situations.*

6.1 Introduction

Recent oil contamination hazards in the open sea (see [215, 271]), shows the importance of finding solutions to remove the oil in an efficient way. To do that, there exist a large number of cleaning technologies [261]. Here, we focus in the use of a Skimmer ship (i.e. pumping ship) to clean the oil contaminated water [9, 260]. More precisely, given a particular oil contamination scenario during a fixed time interval, we are interested in finding an optimal trajectory for this Skimmer ship in order to find an optimum cleaning process.

In order to solve this complex optimization problem, using mathematical and computational methods, we need to model first the evolution of the oil spots concentration

resulting from the combined effects of diffusion, transport (by wind and sea currents) and the action of the Skimmer ship (that implies transport and a reaction phenomena). To do so, we investigate the following issues:

- i) The modeling of the motion of oil spots resulting from the combined effects of diffusion and of transport by wind and sea currents
- ii) The modeling of the physical phenomena associated with the action of the Skimmer ship, assuming that it follows a pre-assigned trajectory.
- iii) The construction of a numerical model for the simulation of the propagation of the oil spots under the pumping.
- iv) The parallelization of the numerical algorithms.

Then, we formulate mathematically our optimization problem. In particular, we need to model the ship trajectory by the use of a continuous function generated by cubic spline interpolation, where the position of the finite number of interpolation points are the optimization variables. The objective function is designed to maximize the amount of oil pumped during the fixed time interval.

Since this optimization problem seems to have various local and global minima (see Figure 6.7), we solve it by considering the GMSA described previously in Section 2.3.3. We perform a sensitivity analysis of the algorithm parameters and operators, to choose the most suited to our problem.

To verify the efficiency of our approach, we consider and solve numerically three particular examples for various oil spots shapes based on real cases [215].

In Sections 6.2-6.5, we introduce the numerical model considered here to simulate the movement of the oil spots and the effect of the Skimmer ship. Then, in Section 6.6, we perform various experiments to analyze some characteristics (i.e. stability, precision and computational time) of this model. Finally, Section 6.7 presents the optimal trajectory problem. In particular, we show the numerical results over three particular examples.

6.2 A mathematical model for oil spills

In this Chapter, we address the pollution cleaning problem assuming information on the wind and sea currents velocity fields in a time interval $(0, T)$. We consider a spatial domain $\Omega = (x_{1,\min}, x_{1,\max}) \times (x_{2,\min}, x_{2,\max}) \subset \mathbb{R}^2$, large enough to ensure that the pollutant will stay in Ω during the corresponding time interval. We assume for simplicity that the density of the pollutant is smaller than the one of the sea water (so that it remains at the top) and the layer-thickness of the pollutant is a constant h . The office of response and restoration of the U.S. National Ocean Service (see <http://response.restoration.noaa.gov>) provides the estimation of the values of h , given in Table 6.1, depending on the color of the oil in the water. We denote by $c(x, t)$ the pollutant superficial concentration, measured as the volume of pollutant per surface area at $\{x, t\} \in \Omega \times (0, T)$. We assume that in the absence of pumping, the evolution of c is governed by three main effects, namely:

Oil Color	Layer-Thickness Interval (μm)
Silver	0.04 - 0.30
Rainbow	0.30 - 5.0
Metallic	5.0 - 50
Transitional Dark	50 - 200
Dark	≥ 200

Table 6.1: Layer-thickness interval of the oil spot in metric unit for each of the oil codes. Data from the office of response and restoration of the U.S. National Ocean Service: <http://response.restoration.noaa.gov>

- (i) Diffusion of the pollutant
- (ii) Transport due to the wind
- (iii) Transport due to the sea currents

Under these assumptions, the space-time distribution of c is governed by the following advection-diffusion type equation:

$$\begin{cases} \frac{\partial c}{\partial t} + \nabla \cdot \mathbf{J} = 0 & \text{in } \Omega \times (0, T), \\ c = 0 & \text{on } \partial\Omega \times (0, T), \\ c(x, 0) = c_0(x), & x \in \Omega. \end{cases} \quad (6.2.1)$$

In (6.2.1), $c_0(\cdot)$ is the initial superficial concentration in Ω and the flux \mathbf{J} can be decomposed as

$$\mathbf{J} = \mathbf{J}_d + \mathbf{J}_w + \mathbf{J}_s, \quad (6.2.2)$$

where

- \mathbf{J}_d is the diffusion flux, given by

$$\mathbf{J}_d = -\mathbf{k}\nabla c, \quad (6.2.3)$$

with $\mathbf{k} = \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix}$ and k_1, k_2 two positive constants.

- \mathbf{J}_w is the wind flux, given by

$$\mathbf{J}_w = c \mathbf{w}, \quad (6.2.4)$$

where \mathbf{w} is the wind velocity multiplied by a suitable drag factor.

- \mathbf{J}_s is the flux associated with the transport due to sea currents, given by

$$\mathbf{J}_s = c \mathbf{s}, \quad (6.2.5)$$

where \mathbf{s} is the current velocity.

Combining (6.2.1) with (6.2.2)–(6.2.5), we obtain

$$\begin{cases} \frac{\partial c}{\partial t} - \nabla \cdot \mathbf{k} \nabla c + \nabla \cdot c \mathbf{w} + \nabla \cdot c \mathbf{s} = 0 & \text{in } \Omega \times (0, T), \\ c = 0 & \text{on } \partial\Omega \times (0, T), \\ c(x, 0) = c_0(x), & x \in \Omega. \end{cases} \quad (6.2.6)$$

Remark 12 *The homogeneous boundary condition in (6.2.1) and (6.2.6) follows from that fact that Ω has been assumed large enough to contain the oil spills during the time interval $(0, T)$ (of course this assumes that \mathbf{k} is small enough so that the positive values of c on $\partial\Omega$ resulting from diffusion can be neglected). In a forthcoming publication we will replace the linear diffusion term in (6.2.6) by a nonlinear one producing a diffusion propagating with finite velocity.*

Remark 13 *The Coriolis effect, is actually included in the sea current \mathbf{J}_s .*

6.3 Modelling of the pumping process

Among the various techniques which have been employed to remediate oil spills related pollution, we will focus on the one based on a pumping process carried on by a Skimmer ship (see <http://apgdepollution.free.fr> for details). From a modeling point of view an important step is the inclusion of the reacting effect in system (6.3.1). Concerning the pumping process, we are going to assume that:

- The Skimmer ship follows a pre-assigned trajectory $\gamma_p(t), t \in [0, T]$, that remains inside the region Ω .
- The pump is a cylinder with a cross section of radius R_p and height h_p (we suppose $h_p \geq h$) that pumps the fluid at a velocity Q in the radial directions. Therefore, the pumped oil volume per unit time is $2\pi R_p Q c$ corresponding to a pump density of $\frac{2\pi R_p Q c}{\pi R_p^2}$.

From the above assumptions, the variant of (6.2.1) including the pumping effect reads as follows:

$$\begin{cases} \frac{\partial c}{\partial t} - \nabla \cdot \mathbf{k} \nabla c + \nabla \cdot c \mathbf{w} + \nabla \cdot c \mathbf{s} \\ \quad + \nabla \cdot c \mathbf{p} = -\frac{2Q}{R_p} c \chi_{B(\gamma_p(t), R_p)}(x), & \text{in } \Omega \times (0, T), \\ c = 0, & \text{on } \partial\Omega \times (0, T), \\ c(x, 0) = c_0(x), & x \in \Omega, \end{cases} \quad (6.3.1)$$

with

$$\mathbf{p} = \begin{cases} QR_p \frac{\overrightarrow{\gamma_p(t)x}}{|\overrightarrow{\gamma_p(t)x}|^2}, & \forall x \in \Omega \setminus \bar{B}(\gamma_p(t), R_p), \\ 0, & \forall x \in B(\gamma_p(t), R_p), \end{cases} \quad (6.3.2)$$

and

$$\chi_{B(\gamma_p(t), R_p)}(x) = \begin{cases} 0, & \forall x \in \Omega \setminus \bar{B}(\gamma_p(t), R_p), \\ 1, & \forall x \in B(\gamma_p(t), R_p), \end{cases} \quad (6.3.3)$$

where $\bar{B}(\gamma_p(t), R_p)$ is the ball of center $\gamma_p(t)$ and radius R_p .

Remark 14 *System (6.3.1) is an advection-reaction-diffusion problem, the reaction term being associated with the right hand side of the first equation in (6.3.1) and with (6.3.2).*

Remark 15 *We could approximate (6.3.2) by*

$$\mathbf{p} = \psi(|\overrightarrow{\gamma_p(t)x}|) \frac{\overrightarrow{\gamma_p(t)x}}{|\overrightarrow{\gamma_p(t)x}|^2} \quad (6.3.4)$$

with

$$\psi(r) = \begin{cases} 0, & \text{if } 0 \leq r < R_p, \\ \frac{QR_p}{r} & \text{if } R_p \leq r \leq \bar{R}, \\ 0, & \text{if } r > \bar{R}, \end{cases} \quad (6.3.5)$$

where \bar{R} is large enough to ensure that the velocity field generated by the pump, at a distance \bar{R} of the pump, can be neglected.

Remark 16 *If the wind and the sea currents velocity field $\mathbf{w} + \mathbf{s}$ depend only on time we can perform a change of variables $(t, x) \rightarrow (t, \bar{x})$ where $\bar{x}(t, x) = x + \int_0^t (\mathbf{w}(z) + \mathbf{s}(z)) dz$. This transforms the first equation of (6.3.1) into a new one without the terms with \mathbf{w} and \mathbf{s} on a domain Ω that is moving in time, in the (t, x) variable, with the vector $\int_0^t (\mathbf{w}(z) + \mathbf{s}(z)) dz$. This is interesting in order to reduce the size of the domain to be considered, since in the new problem, we only need to compute the effects of the diffusion and the transport/reaction due to the pump.*

6.4 Numerical approximation

The Finite Volume method is well suited for the space-time discretization of problem (6.3.1).

For the numerical approximation of (6.3.1), given $I, J \in \mathbb{N}$ we divide the spatial domain $\Omega = (x_{1,\min}, x_{1,\max}) \times (x_{2,\min}, x_{2,\max})$ into control volumes $\Omega_{i,j}$. For $i = 1, \dots, I$; $j = 1, \dots, J$, we define

$$\Omega_{i,j} = (x_{1,\min} + (i-1)\Delta x_1, x_{1,\min} + i\Delta x_1) \times (x_{2,\min} + (j-1)\Delta x_2, x_{2,\min} + j\Delta x_2), \quad (6.4.1)$$

with $\Delta x_1 = \frac{x_{1,\max} - x_{1,\min}}{I}$, $\Delta x_2 = \frac{x_{2,\max} - x_{2,\min}}{J}$. We define $\Delta t = \frac{T}{N}$, where $N \in \mathbb{N}$ is the number of time steps.

Considering a fully implicit time discretization of the backward Euler type for the time discretization of (6.3.1) and an upwind scheme for the transport term, one obtains at $t = n\Delta t$ on the cell $\Omega_{i,j}$, for $i = 1, \dots, I$ and $j = 1, \dots, J$, the following scheme:

$$C_{i,j}^0 = C_0(\xi_{i,j}), \quad \xi_{i,j} \text{ being the center of cell } \Omega_{i,j}; \quad (6.4.2)$$

for $n \geq 0$ we compute $\{C_{i,j}^n\}$ from $\{C_{i,j}^{n-1}\}$ using :

$$\begin{aligned} & \frac{C_{i,j}^n - C_{i,j}^{n-1}}{\Delta t} + 2 \left(\frac{k_1}{(\Delta x_1)^2} + \frac{k_2}{(\Delta x_2)^2} \right) C_{i,j}^n \\ & - \frac{k_1}{(\Delta x_1)^2} (C_{i+1,j}^n + C_{i-1,j}^n) - \frac{k_2}{(\Delta x_2)^2} (C_{i,j+1}^n + C_{i,j-1}^n) \\ & + \frac{1}{\Delta x_1} [\max(0, V_{1,i,j-\frac{1}{2}}^n) C_{i,j}^n + \min(0, V_{1,i,j-\frac{1}{2}}^n) C_{i+1,j}^n \\ & \quad - \max(0, V_{1,i-1,j-\frac{1}{2}}^n) C_{i-1,j}^n - \min(0, V_{1,i-1,j-\frac{1}{2}}^n) C_{i,j}^n] \\ & + \frac{1}{\Delta x_2} [\max(0, V_{2,i-\frac{1}{2},j}^n) C_{i,j}^n + \min(0, V_{2,i-\frac{1}{2},j}^n) C_{i,j+1}^n \\ & \quad - \max(0, V_{2,i-\frac{1}{2},j-1}^n) C_{i,j-1}^n - \min(0, V_{2,i-\frac{1}{2},j-1}^n) C_{i,j}^n] \\ & + \frac{2\pi R_p Q}{\Delta x_1 \Delta x_2} C_{i_p,j_p}^n \chi_{i,j}^{p,n} = 0, \end{aligned} \quad (6.4.3)$$

where in (6.4.3)

- $C_{k,l}^n = 0$ if $k = 0$ or $I + 1$ (respectively, $l = 0$ or $J + 1$),
- $\Omega_{i_p,n,j_p,n}$ is the cell containing $\gamma_p(n\Delta t)$ and $\chi_{i,j}^{p,n} = 0$ if $\{i, j\} \neq \{i_p,n, j_p,n\}$, $\chi_{i,j}^{p,n} = 1$ if $\{i, j\} = \{i_p,n, j_p,n\}$,
- $\mathbf{V}(x, t) = (V_1(x, t), V_2(x, t)) = \mathbf{w}(x, t) + \mathbf{s}(x, t) + \mathbf{p}(x, t)$, where $x \in \Omega$ and $t \in [0, T]$.
- $V_{1,i,j-\frac{1}{2}}^n = V_1((x_{1,\min} + i\Delta x_1, x_{2,\min} + (j - \frac{1}{2})\Delta x_2), n\Delta t)$,
- $V_{2,i-\frac{1}{2},j}^n = V_2((x_{1,\min} + (i - \frac{1}{2})\Delta x_1, x_{2,\min} + j\Delta x_2), n\Delta t)$.

System (6.4.2)-(6.4.3) can be rewritten in the form:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad \text{with } \mathbf{x} = \mathbf{C}^n, \quad \mathbf{b} = \mathbf{C}^{n-1}, \quad (6.4.4)$$

where \mathbf{A} is a $IJ \times IJ$ -matrix and \mathbf{C}^n and \mathbf{C}^{n-1} are vectors of dimension IJ corresponding to the concentration matrices $(C_{i,j}^n)$ and $(C_{i,j}^{n-1})$ stored column wise.

The solution of the linear system (6.4.4) is obtained by using a Bi-Conjugate gradient type algorithm (Bi-CG), well suited for non-symmetric matrices such as \mathbf{A} . A particular parallel implementation of this algorithm is presented in Section 6.5.

Remark 17 : *In Section 6.6.3, we will discuss the advantages of using the implicit instead of explicit schemes.*

Remark 18 : *To space discretize the advecting term $\nabla \cdot c\mathbf{V}$ we have employed a first order upwinding scheme.*

Remark 19 : *For a thorough discussion of finite volume methods for the solution of partial differential equations see [73].*

6.5 Parallel Bi-Conjugate Gradient type method

In order to choose the numerical method appropriate to solve system (6.4.4), we have taken into account the fact that A is a non-symmetric matrix, that the sea region under consideration may be quite large and thus that the space discretization might produce a large dimensional problem that requires large storage space, and that the time step necessary for stability might increase the computational time.

The generalized minimal residual method GMRES, is a suitable method for nonsymmetric systems, but it generates long recurrences in order to keep all residuals orthogonal, at the cost of increasing storage demand.

The biconjugate gradient method (Bi-CG) first developed by Lanczos [163] and further studied by Fletcher [89], takes another approach, replacing the orthogonal sequence of residuals by two mutually orthogonal sequences, at the price of no longer providing a minimization of the residuals as the GMRES. However, it has theoretical properties similar to the CG method, and returns the exact solution in at most IJ iterations (for exact computations) and can be seen as a direct method. If A is symmetric, it produces the same iterates x_k as CG but at twice the computational cost.

In terms of number of iterations, the Bi-CG method is comparable to GMRES [95] and attains similar accuracy. The generation of the basis for Bi-CG is cheap and thus it requires low memory. Considering that Bi-CG needs to compute two matrix-vector multiplications (instead of one with GMRES), an efficient way to perform the matrix-vector products has to be designed to make the method efficient. (See also ([276])).

Taking into account the above considerations, we have chosen the Bi-CG method to use in our problem, and to make it efficient we have implemented on a Distributed Parallel computer. In fact, to get smooth convergence and better accuracy, a stabilization method called Bi-CGSTAB developed by Van der Vost [262, 249], was employed.

6.5.1 Bi-CG and Bi-CGSTAB methods

The induction relations used to update the residuals are constructed so that the original residuals $\mathbf{r}_j = \mathbf{b} - A\mathbf{x}_j$ are bi-orthogonal with respect to the residuals of another system $A^T \tilde{\mathbf{x}} = \tilde{\mathbf{b}}$, $\tilde{\mathbf{r}}_j = \tilde{\mathbf{b}} - A^T \tilde{\mathbf{x}}_j$. This bi-orthogonality of the residuals can be accomplished by two 3-term update or induction relations. There are also relations to update \mathbf{x}_j and $\tilde{\mathbf{x}}_j$ and the new directions \mathbf{p}_j and $\tilde{\mathbf{p}}_j$, which are also orthogonal to the residuals, $\tilde{\mathbf{p}}_i^T \mathbf{r}_j = 0$ and $\tilde{\mathbf{r}}_j^T \mathbf{p}_k = 0$.

The two directions are conjugate, that is $\tilde{\mathbf{p}}_i^T A \mathbf{p}_j = 0$, $i \neq j$ and the residuals are orthogonal, that is $\tilde{\mathbf{r}}_j^T \mathbf{r}_j = 0$, $i \neq j$. Choosing $\mathbf{p}_0 = \mathbf{r}_0$ and $\tilde{\mathbf{p}}_0 = \tilde{\mathbf{r}}_0$, the method terminates in at most IJ steps.

To get smooth convergence, better precision and a faster method, the Bi-CGSTAB method was used. We derive it now.

The residuals can be written as $\mathbf{r}_j = P_j(A) \mathbf{r}_0$ and $\tilde{\mathbf{r}}_j = P_j(A^T) \tilde{\mathbf{r}}_0$. The polynomial $P_j(A)$ should reduce \mathbf{r}_0 , (that depends on the initial \mathbf{x}_0), but it might not reduce any other vector, including $P_i(A) \mathbf{r}_0$, which is needed to obtain

$$(P_i(A) \mathbf{r}_0, P_j(A^T) \tilde{\mathbf{r}}_0) = (P_j(A) P_i(A) \mathbf{r}_0, \tilde{\mathbf{r}}_0) = 0, \quad \text{for } i < j.$$

To avoid this possible irregularity, the residuals can be written as

$$\mathbf{r}_j = Q_j(A) P_j(A) \mathbf{r}_0 \quad \text{where} \quad Q_i(x) = (1 - \omega_1 x)(1 - \omega_2 x) \dots (1 - \omega_i x)$$

Constants ω_j are chosen to minimize the residuals \mathbf{r}_j , in the j -iteration. Due to the orthogonality property $(P_i(A) \mathbf{r}_0, Q_j(A^T) \tilde{\mathbf{r}}_0) = 0$, if $j < i$, we get finite termination in at most IJ steps. The Bi-CGSTAB algorithm follows:

- given \mathbf{x}_0 (the initial approximation), compute $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$;
- take an arbitrary vector $\tilde{\mathbf{r}}_0$ such that $(\tilde{\mathbf{r}}_0, \mathbf{r}_0) \neq 0$, like $\tilde{\mathbf{r}}_0 = \mathbf{r}_0$;
- take $\rho_0 = \alpha = \omega_0 = 1$ and $\mathbf{v}_0 = \mathbf{p}_0 = 0$;

for $i=1$ until convergence

- $\rho_i = (\tilde{\mathbf{r}}_0, \mathbf{r}_{i-1})$;
- $\beta = (\frac{\rho_i}{\rho_{i-1}})(\frac{\alpha}{\omega_{i-1}})$;
- $\mathbf{p}_i = \mathbf{r}_{i-1} + \beta(\mathbf{p}_{i-1} - \omega_{i-1}\mathbf{v}_{i-1})$;
- $\mathbf{v}_i = A\mathbf{p}_i$;
- $\alpha = \frac{\rho_i}{(\tilde{\mathbf{r}}_0, \mathbf{v}_i)}$;
- $\mathbf{s} = \mathbf{r}_{i-1} - \alpha\mathbf{v}_i$;
- norm = (\mathbf{s}, \mathbf{s}) ;

If norm $\leq tol1$ set $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i\mathbf{p}_i$, stop ;

- $\mathbf{t} = A\mathbf{s}$;

- $\omega_i = \frac{(\mathbf{t}, \mathbf{s})}{(\mathbf{t}, \mathbf{t})}$;

- $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i\mathbf{p}_i + \omega_i\mathbf{s}$;

If $\|x_i - x_{i-1}\| \leq tol2$, stop ;

- set $\tilde{\mathbf{r}}_i = \mathbf{s} - \omega_i\mathbf{t}$;

end for

- \mathbf{x}_i is the approximation to the solution.

In this algorithm two stopping conditions have been used, with two tolerances $tol1$ and $tol2$ that depend on the desired accuracy for the solution.

With Bi-CGSTAB, besides the two matrix vector products also needed in Bi-CG, four inner products have to be computed, instead of two as in the original method. However, this extra work will be compensated by the decrease in the number of iterations and the smoother convergence behavior. In addition, four additional IJ -vectors have to be stored $\tilde{\mathbf{r}}_0, \mathbf{p}, \mathbf{v}$ and \mathbf{t} . Some break down situations may occur when either ρ_i or $(\tilde{\mathbf{r}}_0, \mathbf{v}_i)$ become very small, and this situation has to be checked in the algorithm. Eventually, another initial $\tilde{\mathbf{r}}_0$ has to be chosen.

6.5.2 Parallelization of the Bi-CGSTAB

In this Section, we are presenting results from a straight forward parallelization method, testing both row and column wise algorithms to perform the matrix-vector multiplications and the inner products. A more complex parallelization alternative, based on the very recent work of Tong et al. 2009, [107], is now in progress and will be reported in a future work.

In order to parallelize Bi-CGSTAB, we consider the following methodology:

The IJ rows (or columns, depending on the selected parallelization method) of matrix \mathbf{A} are distributed equally to the N_{proc} number of processors. If IJ is not a multiple of N_{proc} , the $N_f = (IJ \text{ modulus } IJ_{\text{proc}})$ remaining rows will be sent to the first N_f processors. We use a coordinate system to store the structured matrix \mathbf{A} .

To generate the initial \mathbf{x}_0 for the Bi-CGSTAB algorithm, the polluted areas in the discretized space are assumed to be circular spots with predefined center and radius. The vector $b^0 = C_{i,j}^0$ represents the concentration of oil in the rectangular grid at the initial time. In this case, it will take values 0 or 1 depending on whether the point (i, j) in the grid is polluted or not and every processor will need to update its components of the right hand side $b^n = C_{i,j}^{n-1}$ (the current concentration), at every time step.

To test the best alternative, the computation of the two matrix-vector products (of the form \mathbf{Ax}) needed in the Bi-CGSTAB algorithm, will be performed Column and Row wise.

Column wise method: The matrix vector multiplication can be expressed as a linear combination of the columns \mathbf{A}_i of matrix \mathbf{A} , with the coefficients being the elements \mathbf{x}_i of vector \mathbf{x} . The result of this linear combination, is the sum of vectors $x_i \mathbf{A}_i$ where \mathbf{A}_i is the i -th column of \mathbf{A} . However as the columns are stored in different processors, the computation of the linear combination needs to be done among processors, with communication of the partial sums of the IJ -vectors $x_i \mathbf{A}_i$. The complete sum can be performed using a Fan-in process (see for instance [103]) or the MPI-Allreduce (see for instance [170]) and they had two different computing times. The final vector \mathbf{Ax} will be stored in the root processor and needs to be redistributed for further operations. The length of the vectors to be sent to the processors is IJ .

Row wise method: Each processor has only part of \mathbf{x}_i and the entire vector is needed to compute \mathbf{Ax} . Then, this information has to be communicated to all processors. However, the length of this information depends on the number of rows per processor, and is then much smaller than the one used in the column wise communicated data. Therefore as

the number of processors increases the length of the data to be communicated decreases. This is main reason why this alternative is more efficient. Thus the matrix vector multiplication \mathbf{Ax} is performed in perfect parallelism, where \mathbf{Ax} remains distributed without any further communication.

The four inner products of the Bi-CGSTAB algorithm (of the form (y, z)) are performed using a row wise method and communication is again needed to compute the final sum, and to send the result back to the processors.

The tests performed using both alternatives will be presented in Section 6.6.4, and they show clearly the superiority of the row wise method, which will then be used to perform the numerical simulations presented in Section 6.6.5.

6.6 Numerical analysis of the model

6.6.1 Parameters

In this Section, the model parameters are set in the following way (using the SI unit system):

The modeling domain Ω is defined by $x_{1,\min} = 0$, $x_{1,\max} = 2 \times 10^4$, $x_{2,\min} = 0$ and $x_{2,\max} = 2 \times 10^4$.

The simulation time is equal to one day, $T = 86400$.

We consider 4 alternative spatial discretization meshes (I, J) : $(50, 50)$, $(100, 100)$, $(200, 200)$ or $(300, 300)$.

The time step is $\Delta t = 86.4$ (i.e. $N = 1000$). This choice is discussed in Section 6.6.3.

The diffusion coefficients are $k_1 = k_2 = 0.5$.

The wind plus sea velocity field $\mathbf{s}(x, t) + \mathbf{w}(x, t)$ is defined by

$$\left(\frac{x_1}{4x_{1,\max}} \cos\left(\frac{\pi t}{3600}\right), \frac{x_2}{4x_{2,\max}} \sin\left(\frac{\pi t}{3600}\right) \right), \quad (6.6.1)$$

for $t \in [0, T]$ and $x = (x_1, x_2) \in \Omega$.

The pump parameters are $Q = 100$ and $R_p = 1$. We consider two different trajectories of the Skimmer ship:

- Trajectory **T1**: The initial position of the Skimmer ship is $(600, 1400)$ and its trajectory is given by

$$\gamma_p(t) = (x_{1,\max}(0.5 + 0.2 \cos(\frac{\pi t}{86400})), x_{2,\max}(0.3 + 0.1 \sin(\frac{\pi t}{86400}))). \quad (6.6.2)$$

- Trajectory **T2**: The initial position of the Skimmer ship is $(800, 1400)$ and its trajectory is given by

$$\gamma_p(t) = (x_{1,\max}(0.5 + 0.2 \cos(\frac{\pi t}{21600})), x_{2,\max}(0.4 + 0.1 \sin(\frac{\pi t}{86400}))). \quad (6.6.3)$$

The two Skimmer ship trajectories **T1** and **T2** are depicted in Figures 6.1 and 6.2.

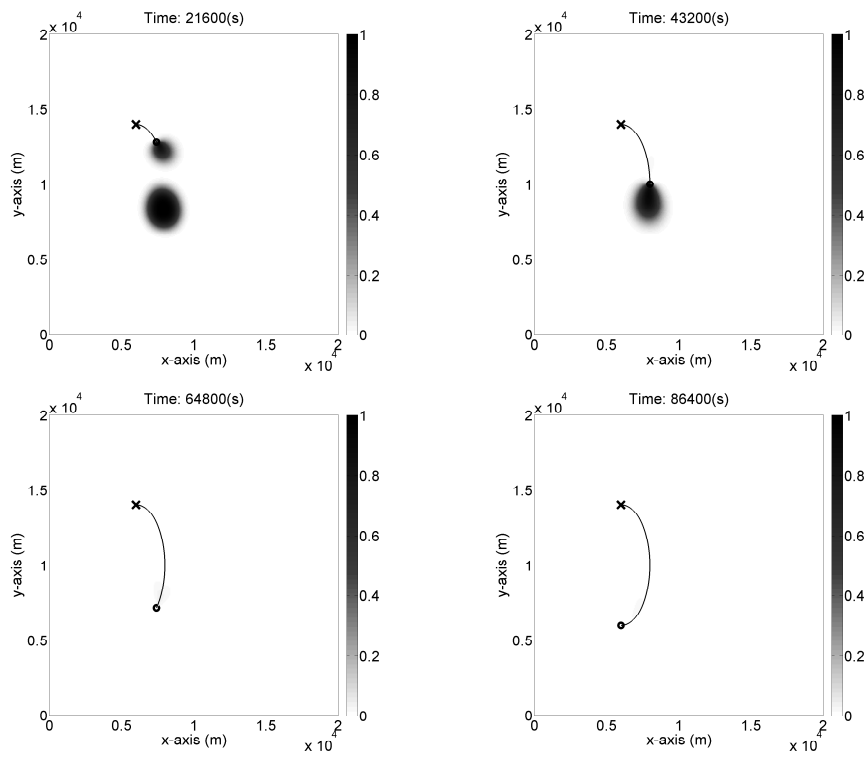


Figure 6.1: The evolution of the concentration considering the trajectory **T1** for the mesh 300×300 at times (**Top-Left**) 21600 s, (**Top-Right**) 43200 s, (**Bottom-Left**) 64800 s and (**Bottom-Right**) 86400 s. The initial position (X), current position (o) and trajectory (—) of the pump are also shown.

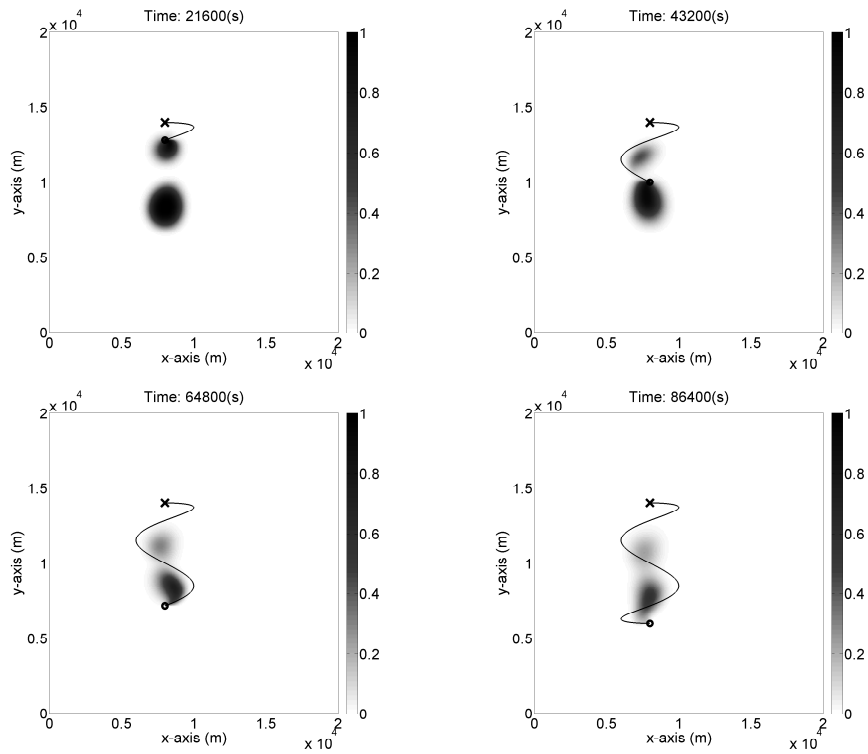


Figure 6.2: The evolution of the concentration considering the trajectory **T2** for the mesh 300×300 at times (**Top-Left**) 21600 s, (**Top-Right**) 43200 s, (**Bottom-Left**) 64800 s and (**Bottom-Right**) 86400 s. The initial position (**X**), current position (**o**) and trajectory (—) of the pump are also shown.

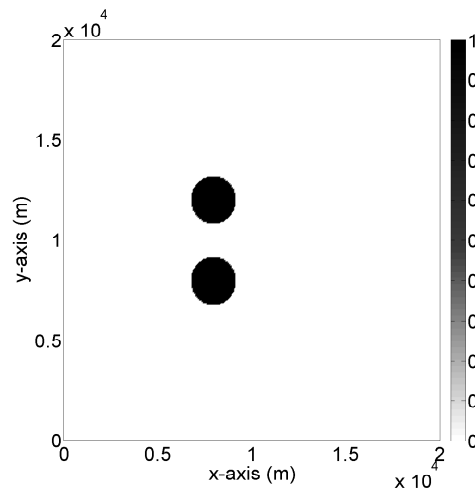


Figure 6.3: Initial position of the pollutant spots (in black) in the domain Ω for the 300×300 mesh.

The initial pollutant concentration, presented in Figure 6.3, is given by

$$c(x, 0) = \chi_{B((8000,8000),1200)}(x) + \chi_{B((8000,12000),1200)}(x), \quad (6.6.4)$$

where $\chi_{B(b,a)}$ is defined in (6.3.3).

The stopping conditions used for the Bi-CGSTAB Method is $tol1 = tol2 = 10^{-6}$.

For all computations, we have used the **BlueGene/L System** from the **Dares -bury Laboratory** in England. We have used up to $N_{\text{proc}} = 32$ bi-cores processors. Each processor has a 32-Bit architecture and 1 GB of local memory. The code is programmed in Fortran 90 and uses the MPI standard protocol. Double precision values were used in all computations. All runs were performed using the row wise method. For the mesh 200×200 we have also tested the Fan-in and reduced column wise methods to illustrate the comparison between the parallelization methods.

6.6.2 Results

In this Section, we first present a comparison between the implicit scheme (6.4.3) and its corresponding fully explicit version. Then we report the results related to the parallelization schemes. Finally, we analyze and discuss the behavior of our model when modifying the time and space discretization steps.

6.6.3 Comparison between implicit and explicit schemes

In order to check the efficiency of the implicit scheme (6.4.3) and the corresponding explicit one, we compare their stability. To do this, we consider the trajectory **T2** (similar results have been obtained with **T1**), the meshes 50×50 , 100×100 , 200×200 and 300×300 and the time steps $\Delta t = 8640, 1728, 864, 345.6, 172.8, 115.2, 86.4, 17.28$, and 8.64 (i.e. $N = 10, 50, 100, 250, 500, 750, 1000, 5000$ and 10000 , respectively). The quantity of pollutant pumped during the simulation processes are shown in Table 6.2 for the implicit scheme.

When $N \geq 250$, the implicit scheme was stable. Furthermore, the time step $\Delta t = 86.4$ (i.e. $N = 1000$) gives a good compromise between computational complexity and precision. Indeed, the solutions obtained with this time step and $\Delta t = 8.64$ (i.e. $N = 10000$) exhibit a difference on the quantity of pollutant pumped inferior to 0.6%, which is quite acceptable. This time step value is used in all computations in Sections 6.6.4 and 6.6.5.

On the other hand, the explicit scheme is always unstable for those values of the discretization. It is well known that the explicit schemes demand time step restrictions of CFL type (see [100, 126]). More precisely, for all $t \in [0, T]$ and $x \in \Omega$, Δt should satisfy

$$\Delta t < \frac{\Delta x_1 \Delta x_2}{\Delta x_2 |\mathbf{V}_1(x, t)| + \Delta x_1 |\mathbf{V}_2(x, t)|}. \quad (6.6.5)$$

In the considered numerical tests, for all $t \in [0, T]$ and $x \in \Omega$ we have

$$|\mathbf{V}_1(x, t)| + |\mathbf{V}_2(x, t)| < (Q + \frac{1}{4})(\sin(\frac{\pi}{4}) + \cos(\frac{\pi}{4})) < 142. \quad (6.6.6)$$

Hence, we should set $\Delta t < 2.80, 1.40, 0.70$ and 0.46 (i.e. $N > 30858, 61715, 123429$ and 187827 , respectively), when using the $50 \times 50, 100 \times 100, 200 \times 200$ and 300×300 meshes, respectively.

Thus, the explicit scheme needs time steps much smaller than the implicit one, especially for the most refined meshes, and it is then less interesting from a computational point of view.

Remark 20 *The reaction term (i.e. $-(2Qc\chi_{B(\gamma_p(t), R_p)}(x))/R_p$) added on the right hand side of the first equation of (6.3.1) also impose a restriction on Δt when considering the explicit scheme. Indeed, the explicit discrete version of this term, i.e. $(2\pi R_p Q C_{i_p, j_p}^{m-1} \chi_{i, j}^p)/(\Delta x_1 \Delta x_2)$ (using the same notations as in (6.4.3)), needs time steps satisfying $\Delta t < (\Delta x_1 \Delta x_2)/(2\pi R_p Q)$ in order to ensure the positivity of the solution and the stability of the scheme. More precisely, we must set $\Delta t < 254, 63, 15$ and 7 (i.e. $N > 341, 1371, 5760$ and 12343 , respectively) when considering the $50 \times 50, 100 \times 100, 200 \times 200$ and 300×300 meshes, respectively. Similar results have been obtained for other advection-reaction-diffusion equations, see for example [265, 40].*

N	50×50	100×100	200×200	300×300
10	55.37	54.54	unstable	unstable
50	64.68	66.87	64.90	unstable
100	67.15	69.04	68.87	unstable
250	69.74	71.11	70.11	69.99
500	70.32	71.76	70.78	70.40
750	70.75	71.99	70.92	70.50
1000	70.82	72.10	71.00	70.54
5000	71.15	72.39	71.17	70.64
10000	71.2	72.42	71.19	70.65

Table 6.2: Quantity of pollutant pumped, expressed in % with respect to the initial pollutant quantity, observed during the simulation process considering the trajectory **T2**, the meshes $50 \times 50, 100 \times 100, 200 \times 200$ and 300×300 and the time steps associated to the given values of N for the implicit discretization scheme. We also report in this table the unstable results.

6.6.4 Parallelization schemes

As described in Section 6.5.2, the matrix-vector multiplication of the Bi-CGSTAB has been performed using column (combined with Fan-in or All-Reduced summation method) and row wise schemes. The comparison of these methods, when applied to our model, is reported in Table 6.3 and in Figure 6.4 only for trajectory **T1**, results for **T2** being similar.

These results show the advantage of using the row wise scheme, and if column wise is used, they indicate the clear advantage of the All Reduced algorithm. From now on, all results reported were performed in the row wise mode.

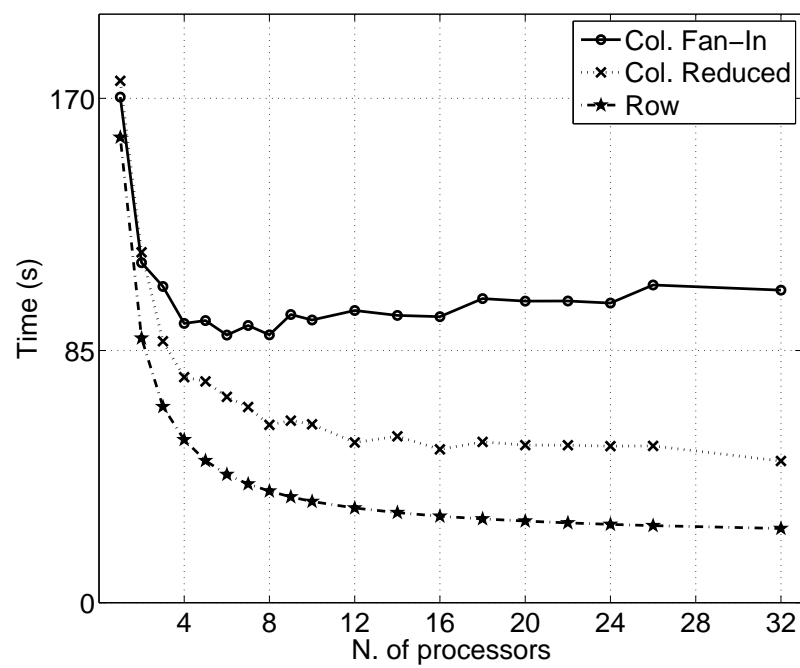


Figure 6.4: Total time, in seconds, using Fan-In (—) and Reduced (..) Column Wise and Row Wise (-.) methods Vs number of processors for mesh 200×200 .

N_{proc}	Fan-In Col.	All-Reduced Col.	Row
1	178.2	178.0	164.0
4	98.4	79.5	57.7
8	94.4	62.7	39.3
16	100.8	54.0	30.4
32	110.1	49.9	26.1

Table 6.3: Time, in seconds, needed to solve the model considering trajectory **T1** and using Fan-In, All-Reduced column and row wise methods in function of N_{proc} for mesh 200×200 .

An important issue is the degree of discretization and the computational time needed to accomplish a given accuracy in the model solution. Therefore, we report and compare the total time, in seconds, needed to perform the simulation in the prefixed time interval (here, one day), for all selected discretization meshes and for $N_{\text{proc}} = 1, 4, 8, 16, 32$. These results are presented in Table 6.4 and in Figure 6.5.

N_{proc}	50×50	100×100	200×200	300×300
1	32.3	78.9	164.0	377.4
4	10.6	27.5	57.7	120.9
8	7.6	19.1	39.3	76.9
16	5.9	15.0	30.4	55.4
32	5.3	12.9	26.1	45.0

Table 6.4: Time, in seconds, needed to solve the model with respect to N_{proc} (line) and the discretization mesh (column).

The computational time is reduced from six up to eight times when using from 1 to 32 processors. As we can observe in Figure 6.5, the largest gain is obtained from 1 to 16 processors.

6.6.5 Influence of the trajectory

In Table 6.5, we report the percentage of the pollutant pumped for the three meshes considered and for T1 and T2. To be able to determine which mesh is the most efficient (in computational time and precision) for the trajectories considered here, we compute the average and maximum difference (in time) between the amount of pumped pollutant obtained with the finest mesh (300×300) and the other meshes. We observe that the accuracy increases with the level of discretization. However, we point out, that the final concentrations obtained with the two finest meshes show similar amount of pumped pollutant (with a mean and maximum difference less than 1.0% and 2.4%, respectively). This makes clear that we need to use fine meshes only up to certain degree (as 200×200 is numerically close to 300×300 and is four times faster).

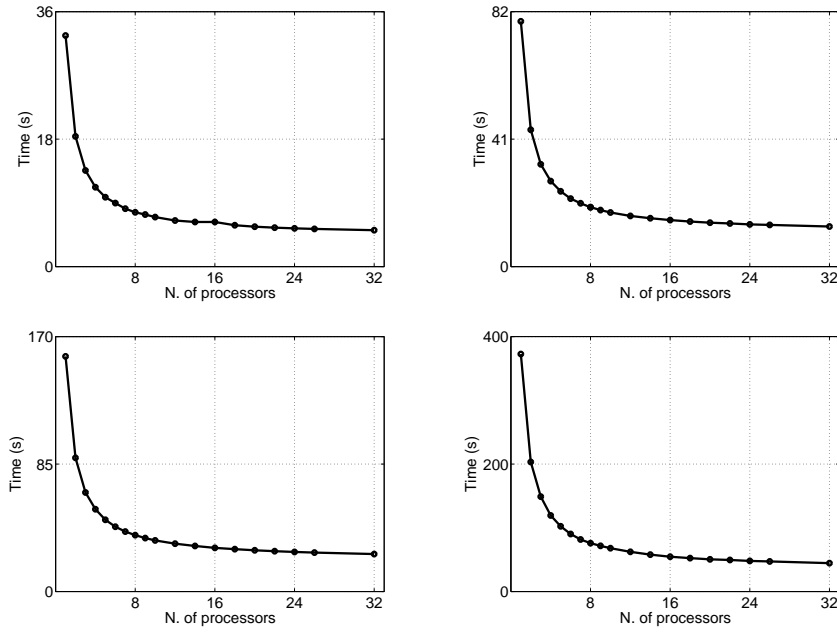


Figure 6.5: Total time (s) vs number of processors for meshes (**Top-Left**) 50×50 , (**Top-Right**) 100×100 , (**Bottom-Left**) 200×200 and (**Bottom-Right**) 300×300 .

Part of the previous differences between the amount of pumped pollutant obtained with the finest mesh and the other meshes, is due to the artificial diffusion of the numerical scheme (6.4.3). This can be observed in Figure 6.6 that depicts the final concentration, for each mesh, considering the trajectory **T2** (results are similar using **T1**). This artificial diffusion phenomena increases with the spatial block size.

In Figures 6.1 to 6.2, we show the evolution of the concentration of the pollutant considering the trajectories **T1** and **T2** at different times for the mesh 300×300 . These figures and Table 6.5 show the impact of the trajectory in the amount of pumped pollutant, pointing out the need to find optimal trajectories to pump a maximum quantity.

6.7 Optimal trajectory

6.7.1 Problem definition

As mentioned in Section 6.1, we address the problem of finding an optimal trajectory for the Skimmer ship, for a particular oil contamination scenario during a fixed time interval $(0, T)$.

For the given time T , we minimize the concentration $c(\xi, T)$ of the remaining pollutant in Ω , which is equivalent to maximize the amount of pumped oil from the sea. More precisely, we are interested in solving the following optimization problem:

$$\min_{\gamma \in D_c} J_c(\gamma) \quad (6.7.1)$$

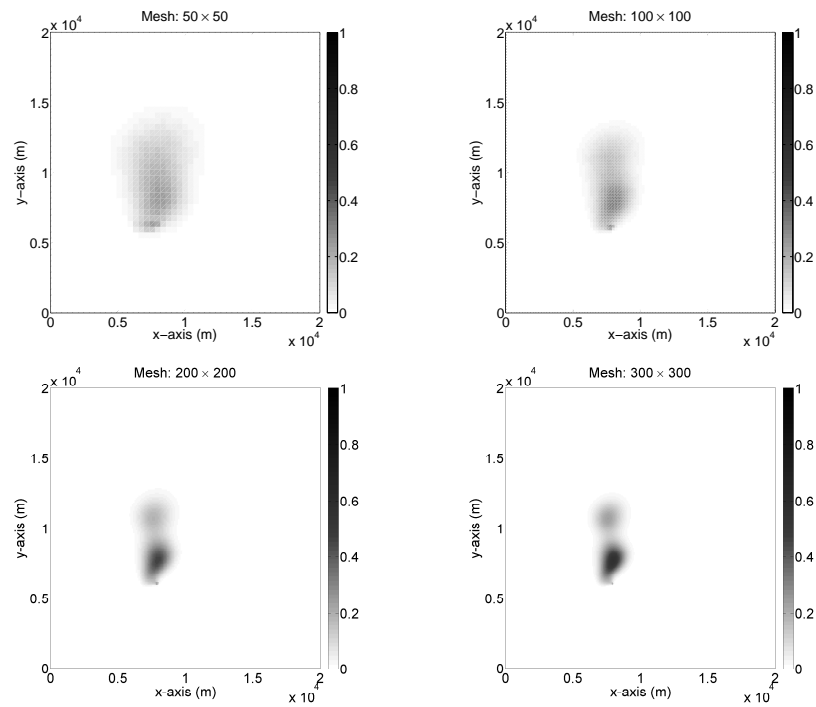


Figure 6.6: Final concentration obtained considering the trajectory **T2** for the meshes (**Top-Left**) 50×50 , (**Top-Right**) 100×100 , (**Bottom-Left**) 200×200 and (**Bottom-Right**) 300×300 .

Traj.	Poll. pump.	50×50	100×100	200×200	300×300
T1	Poll. pump.	84.06	93.01	97.54	98.59
	Mean Diff.	9.99	4.09	0.89	—
	Max Diff.	19.50	9.33	2.32	—
T2	Poll. pump.	70.81	72.09	71.00	70.54
	Mean Diff.	2.27	1.40	0.35	—
	Max Diff.	6.90	3.00	1.20	—

Table 6.5: Quantity of pollutant pumped (Poll. pump.) (expressed in % respecting to the initial pollutant quantity), Mean difference (Mean Diff.) and maximum difference (Max Diff.) (in % with respect to the solution obtained with the mesh 300×300) observed during the simulation considering the trajectories (Traj.) **T1** and **T2** and the meshes 50×50 , 100×100 , 200×200 and 300×300 .

where $J_c(\gamma) = \iint_{\Omega} c(0, x)dx - \int_0^T c(\tau, \gamma(\tau))Qd\tau$ is the objective function, $D_c = \{\gamma \in C^1([0, T], \Omega) \text{ such that } |\gamma'(t)| \leq V_{\max}, \forall t \in [0, T]\}$ is the feasible region and V_{\max} is the maximum velocity of the ship when performing the pumping process. This restriction on the length of γ avoids to consider trajectories implying non realistic ship velocities.

In order to find numerically a smooth optimal pump trajectory (i.e. without sharp corners), we consider trajectories built by using cubic spline interpolation through $n_{\text{npi}} \in \mathbb{N}$ 2-D interpolation points.

The set of interpolation points, denoted by P_{int} , is constructed by using a polar representation:

$$P_{\text{int}} = \{(r_1, \theta_1), \dots, (r_{n_{\text{npi}}}, \theta_{n_{\text{npi}}})\},$$

where $r_i \in [0, r_{\max}]$, with $r_{\max} = V_{\max} * (T/n_{\text{npi}})$ (modeling the ship velocity constraint), and $\theta_i \in [0, 2\pi)$, for $i = 1, \dots, n_{\text{npi}}$.

Given an interpolation point expressed in Cartesian coordinates $(x_k^{\text{int}}, y_k^{\text{int}})$, with $k \in \{1, \dots, n_{\text{npi}} - 1\}$, the next one $(x_{k+1}^{\text{int}}, y_{k+1}^{\text{int}})$ is built as:

$$x_{k+1}^{\text{int}} = x_k^{\text{int}} + r_k \cos(\theta_k),$$

$$y_{k+1}^{\text{int}} = y_k^{\text{int}} + r_k \sin(\theta_k).$$

The resulting interpolated trajectory is denoted by γ or (γ_x, γ_y) or $\gamma_{(r_i, \theta_i)}$.

Furthermore, we need to avoid the ship leaving the domain of study Ω . To accomplish this, we project the trajectory γ using an orthogonal projector on Ω , called Pr_{Ω} , defined as:

$$Pr_{\Omega}(\gamma_{(r_i, \theta_i)}(\tau)) = \begin{pmatrix} \max(\min(\gamma_x(\tau), x_{\max}), x_{\min}), \\ \max(\min(\gamma_y(\tau), y_{\max}), y_{\min}) \end{pmatrix}. \quad (6.7.2)$$

Thus, the numerical optimization problem that we solve, is of the form:

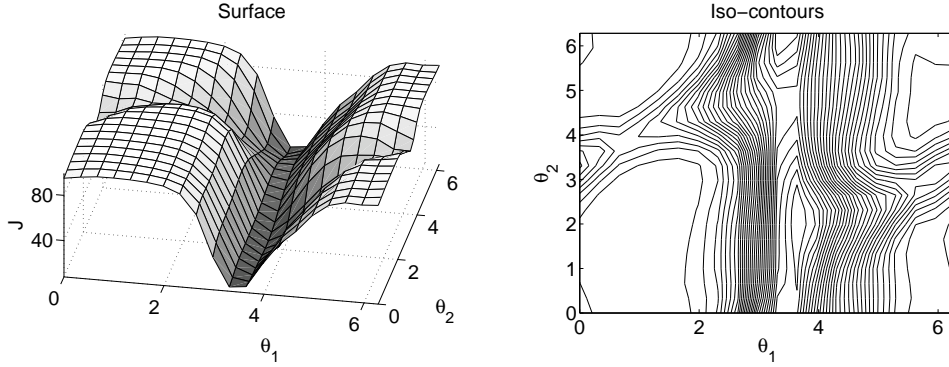


Figure 6.7: 2D graphical representation of function $J(r_i, \theta_i)$ obtained by considering the example 1 presented in Section 6.7.2, with $r_1 = r_2 = 5000$ and two variables θ_1 and θ_2 : (Left) surface and (Right) isocontours.

$$\begin{aligned}
 & \min J(r_i, \theta_i) \\
 & \text{subject to} \\
 & 0 \leq r_i \leq r_{\max}, \quad i = 1, \dots, n_{\text{mpi}}, \\
 & 0 \leq \theta_i < 2\pi, \quad i = 1, \dots, n_{\text{mpi}},
 \end{aligned} \tag{6.7.3}$$

where $J(r_i, \theta_i) = \iint_{\Omega} c(0, x) dx - \int_0^{T_o} c(\tau, \gamma_{(r_i, \theta_i)}(\tau)) Q d\tau$ is the objective function and $\{(r_i, \theta_i)\}_{i=1}^{n_{\text{mpi}}} \subset D$ are the discrete optimization variables with $D = [0, r_{\max}] \times [0, 2\pi)$ is the feasible region. The total number of optimization variables is $N = 2n_{\text{mpi}}$.

Since Problem (6.7.3) has many local and global minima, we need to use a global optimization method capable to find the global solution. A particular 2D graphical representation of function J is shown in Figure 6.7 where we can observe several local minima. Thus, we have applied the GMSA presented in Section 2.3.3 (with B_1 , $t_{l_1} = 30$, $N_g = 40$, $N_p = 40$, $p_c = 0.5$ and $p_m = 0.3$) to solve this problem.

6.7.2 Numerical experiments

In this section, we check the efficiency of our approach. First, we introduce three particular numerical examples. Then, we perform some experiments to determine the optimization algorithm parameters and the most suited operators for our optimization problem. Finally, we present the optimization results obtained for the three cases.

Numerical examples

Due to the fact that in the real world the oil spots have a tendency to adopt various shapes, we have created three representatives examples by considering realistic shape of the oil spots [215] and reasonable (although fictitious) values for the model parameters based on literature [9, 205, 214].

In all cases, the common parameters are the following:

- The computational domain Ω is defined by $x_{\min} = 0$ m, $x_{\max} = 2 \times 10^4$ m, $y_{\min} = 0$ m and $y_{\max} = 2 \times 10^4$ m.
- The constraint $r_{\max} = 1000$ m.
- The number of interpolation points is $n_{\text{mpi}} = 10$.
- The simulation time is equal to one day: $T = 86400$ s.
- The diffusion coefficients are $d_1 = d_2 = 0.5$ m²/s.
- The pump parameters are $Q = 100$ m/s and $R_p = 1$ m.

Furthermore, according to Section 6.6.3, we consider a discretization mesh of $(I, J) = (50, 50)$ and the time step is $\Delta t = 172.8$ s (i.e. $N = 500$). Those values give a good compromises between the model precision and computational time.

The first example has two circular spots defined by:

$$c(\xi, 0) = \chi_{B((8000, 8000), 1200)}(\xi) + \chi_{B((8000, 12000), 1200)}(\xi). \quad (6.7.4)$$

The wind multiplied by a drag factor plus the sea velocity field, $\mathbf{s}(\xi, t) + \mathbf{w}(\xi, t)$, is defined by

$$\left(\frac{x}{4x_{\max}} \cos\left(\frac{\pi t}{3600}\right), \frac{y}{4y_{\max}} \sin\left(\frac{\pi t}{3600}\right) \right), \quad (6.7.5)$$

for $t \in [0, T]$ and $\xi = (x, y) \in \Omega$.

The initial position of the pump is set to $(10000, 14000)$. The initial pollutant concentration and initial position of the pump are depicted in Figure 6.8.

The second example has five ellipsoid spots defined by:

$$\begin{aligned} c(\xi, 0) = & \chi_{E((8000, 8000), 100, 1500)}(\xi) + \chi_{E((12000, 6000), 100, 3000)}(\xi) + \\ & \chi_{E((10000, 10000), 100, 2000)}(\xi) + \chi_{E((6000, 10000), 100, 5000)}(\xi) + \\ & \chi_{E((14000, 6000), 100, 1000)}(\xi) \end{aligned} \quad (6.7.6)$$

where $\chi_{E((a, b), c, d)}(\xi) = 1$ if $(x - a)^2/c + (y - b)^2/d \leq 1$ and 0 elsewhere.

The wind multiplied by a drag factor plus the sea velocity field, $\mathbf{s}(\xi, t) + \mathbf{w}(\xi, t)$, is defined by

$$\left(\frac{1}{7} \cos\left(4\pi \frac{x + (4/7)t}{x_{\max}}\right), 0 \right), \quad (6.7.7)$$

for $t \in [0, T]$ and $\xi = (x, y) \in \Omega$.

The initial position of the pump is set to $(4000, 4000)$. The initial pollutant concentration and initial position of the pump are depicted in Figure 6.8.

The last example has one large spot defined by the following three joint ellipsoids:

$$\begin{aligned} c(\xi, 0) = & \chi_{E((10000, 6000), 1900, 3500)}(\xi) + \chi_{E((12000, 8000), 3000, 1800)}(\xi) + \\ & \chi_{E((12000, 10000), 4000, 2000)}(\xi). \end{aligned} \quad (6.7.8)$$

The wind multiplied by a drag factor plus the sea velocity field, $\mathbf{s}(x, t) + \mathbf{w}(x, t)$, is defined by

$$\left(-\frac{1}{6} \cos\left(10\pi \frac{t}{86400}\right) \frac{x}{x_{\max}}, 0.038 \right), \quad (6.7.9)$$

for $t \in [0, T]$ and $\xi = (x, y) \in \Omega$.

The initial position of the pump is far from the oil spot (to test the ability of the optimal trajectory) and is set to (4000, 14000). The initial pollutant concentration and initial position of the pump are depicted in Figure 6.8.

Results

To perform the numerical experiments described in previous Section, we have used a quad-core computer 64-Bit PC of 2.8Ghz and 12 GB of local memory. The code is programmed in Fortran 90. Double precision values were used in all computations. Each cost function evaluation takes around 1 second.

Furthermore, we have designed a fixed trajectory crossing the initial oil spots at constant velocity, as a reference to compare with the optimal trajectories obtained using the Genetic Algorithm. Those fixed trajectories are depicted in Figure 6.9.

The resulting optimal and fixed a priori trajectories, and their respective final oil concentration distributions, are depicted in Figure 2 (which can be compared with the initial concentrations in Figure 6.8). We point out that in the case of example 1, the gray-scale has been modified in order to emphasize the difference between the concentration distribution for the optimal and for the fixed trajectories.

Furthermore, in Table 6.6, we report the final percentage of pumped oil (objective function value), with respect to the initial concentration, obtained using the optimal and the fixed trajectories.

We can observe in Table 6.6 that the percentage of the remaining oil for the optimal trajectories is sufficiently lower than the fixed ones (the increase of pumped oil is about 10%-15%).

	Example 1	Example 2	Example 3
Fix.	12.34	57.62	69.36
Opt.	1.98	48.61	55.34

Table 6.6: Final percentage of pumped oil obtained with the optimal (Opt.) and fixed (Fix.) trajectories for examples 1,2 and 3.

However we can see that for examples 2 and 3 the remaining quantity of oil on the sea is still important (around 50%). Then for both cases, we have run additional experiments considering that the ship can continue pumping for two days. As the simulated time is twice higher, we have solved the optimization problem with two different number of interpolation points: $n_{\text{npi}} = 10$ (which is the same number as used for one simulation day) and $n_{\text{npi}} = 20$ (which is the double number of points). In order not to change the ship velocity, for 10 interpolation points the maximum distance between two interpolation points is set to $r_{\text{max}} = 2000$ m and for 20 interpolation points $r_{\text{max}} = 1000$ m. Furthermore, according to Section 6.2, the size of domain Ω has been increased but, in order to compare with previous results, the obtained solutions are represented in the initial domain described in Section 6.7.2. Finally, we have also considered the same fixed trajectories than for the case of one simulation day with half velocity.

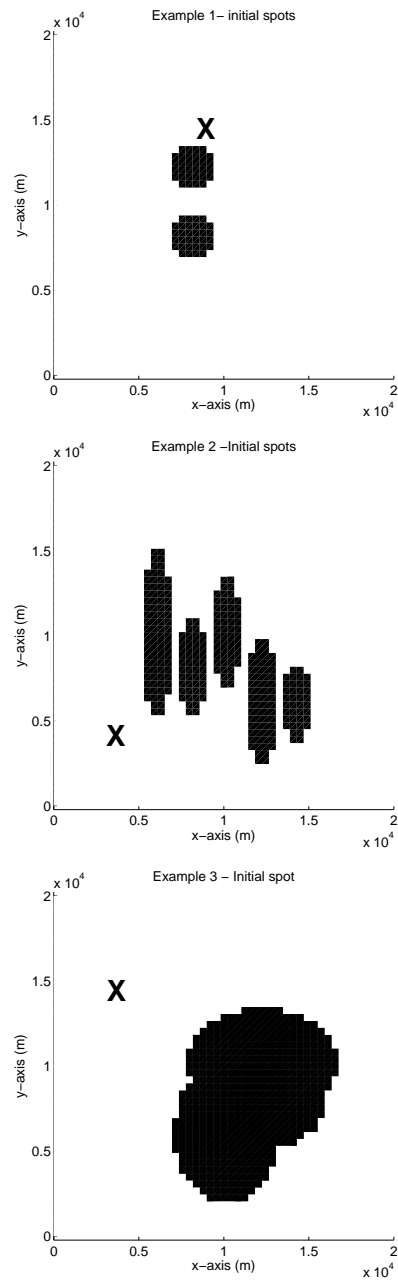


Figure 6.8: Initial position of the pollutant spots (in black the concentration value is 1 whereas in white is 0) in the domain Ω for examples 1 (Top), 2 (Center) and 3 (Bottom). The initial position (X) of the pump is also shown.

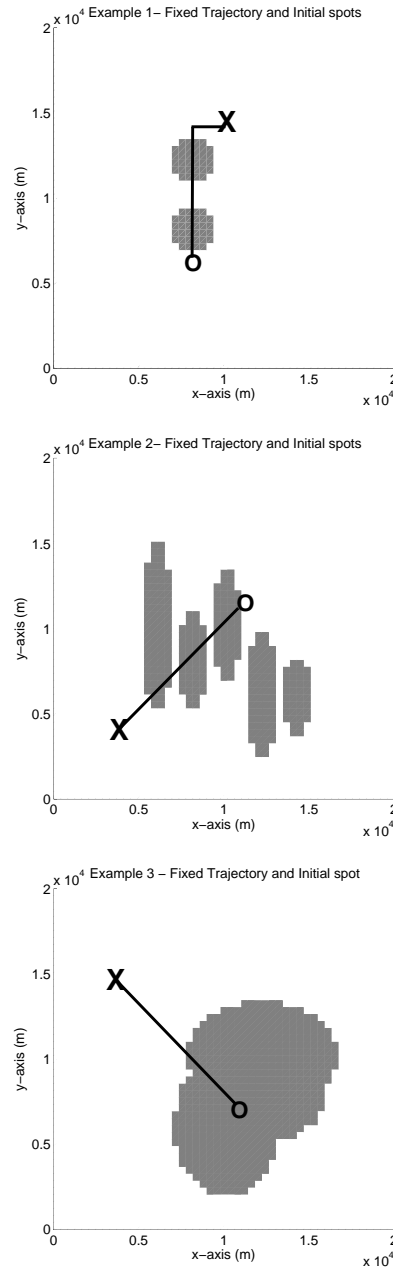


Figure 6.9: Fixed trajectories and the initial spots (in gray the concentration value is 1 whereas in white is 0) for example 1 (Top), 2 (Center) and 3 (Bottom). The initial (X), the final (o) position and the trajectory (—) of the pump are shown.

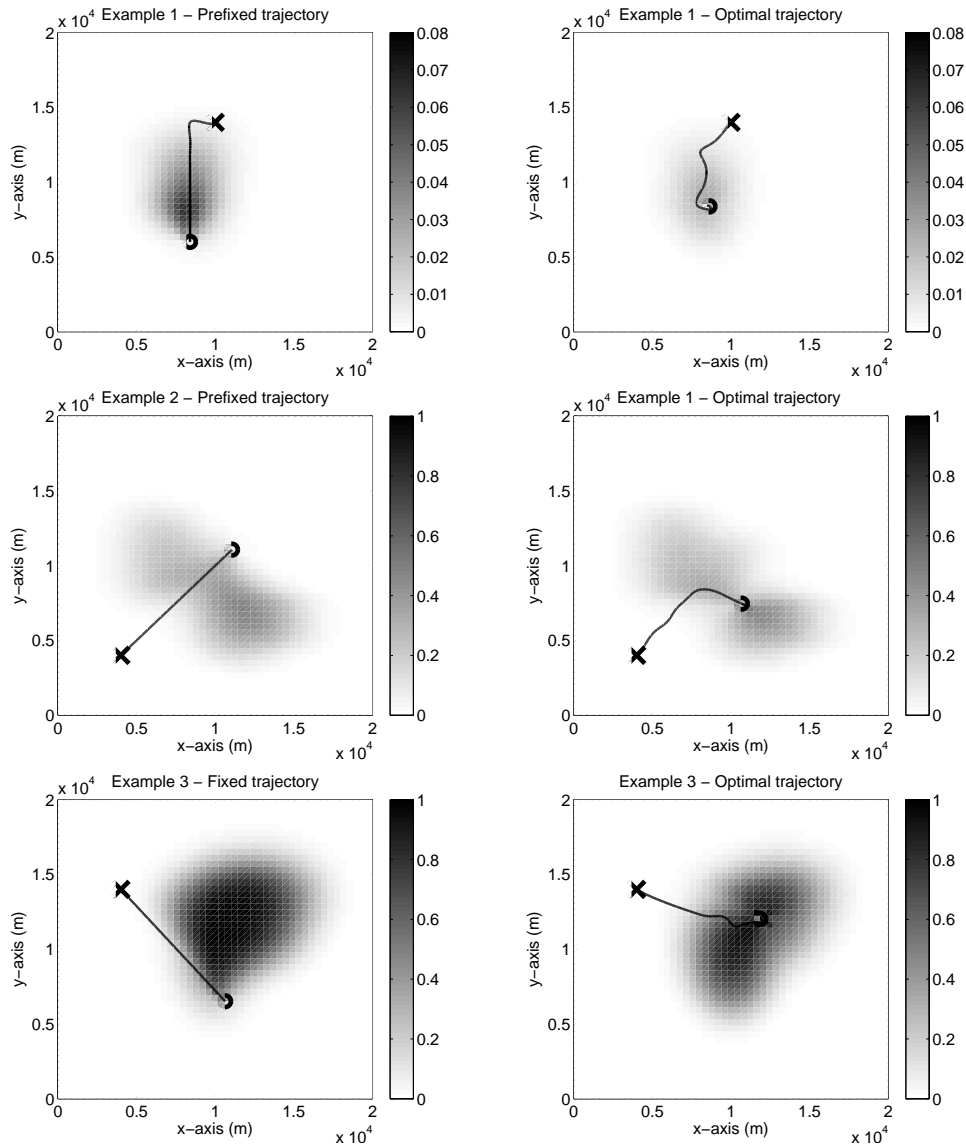


Figure 6.10: Final concentration considering the prefixed (Left) and optimal (Right) trajectories for examples (Top) 1, (Center) 2 and (Bottom) 3. The initial position (X), the final position (o) and trajectory (–) of the pump are also shown. Finally, the prefixed trajectories are presented in the right figures in gray.

For each experiment, the final percentage of pumped oil is presented on Table 6.7. The optimal and fixed trajectories, and their respective final oil concentration, are shown in Figure 6.11 (to be compared with the initial concentrations in Figure 6.8).

	Example 2	Example 3
Fix.	34.30	59.14
$n_{\text{npi}} = 10$	15.95	15.13
$n_{\text{npi}} = 20$	17.39	13.23

Table 6.7: Final percentage of pumped oil obtained with the optimal trajectories, when considering 10 and 20 interpolation points, and the fixed (Fix.) trajectory for examples 2 and 3 and 2 days of simulation.

These results show that increasing the number of simulation days, allows to reduce dramatically the amount of remaining oil in the sea.

As expected, the optimal trajectories are much more efficient than the fixed one designed considering the initial position of the spots.

For the optimization method the increase in the number of interpolation points, which are the number of optimization variables, increases the dimension of the search space. Then, one could expect better results for higher number of interpolation point, as this generates a larger number of possible trajectories. However, since we have used the same number of function evaluation N_{fe} for both cases, it may happen (as for example 2) that the amount of remaining oil is higher than in the case of 10 interpolation points.

The results (trajectories and pumped oil) obtained are similar for both cases of 10 or 20 interpolation points, although the 20 points trajectories are more irregular since the ship increases the number of direction changes (see Figure 6.11). We can conclude that, in these cases, 10 interpolation points are sufficient to achieve a good trajectory.

6.8 Conclusions and Perspectives

In this Chapter, we have presented a two dimensional mathematical model of the motion of oil spots in the open sea, taking into account the velocity and direction of the sea currents, the wind, and the advection and reaction due to a pumping process, carried out by a Skimmer ship with a pre-fixed trajectory.

The implicit and explicit numerical schemes used here to perform the simulation of the model, are finite volume schemes, implemented in parallel on a cluster with up to 32 processors. The results obtained, show clearly that the row wise method used to solve the linear systems associated with the discrete model, is highly superior to the column wise method. Also, the efficiency obtained in the parallelization is improved when increasing the number of processors, showing the parallel code we developed has the expected scaling properties. Furthermore, they seem to indicate to use an implicit scheme to get a computationally efficient algorithm.

Then, we have modeled the Skimmer ship trajectory considering cubic spline interpolation techniques. Those interpolation points are used as the independent variables for

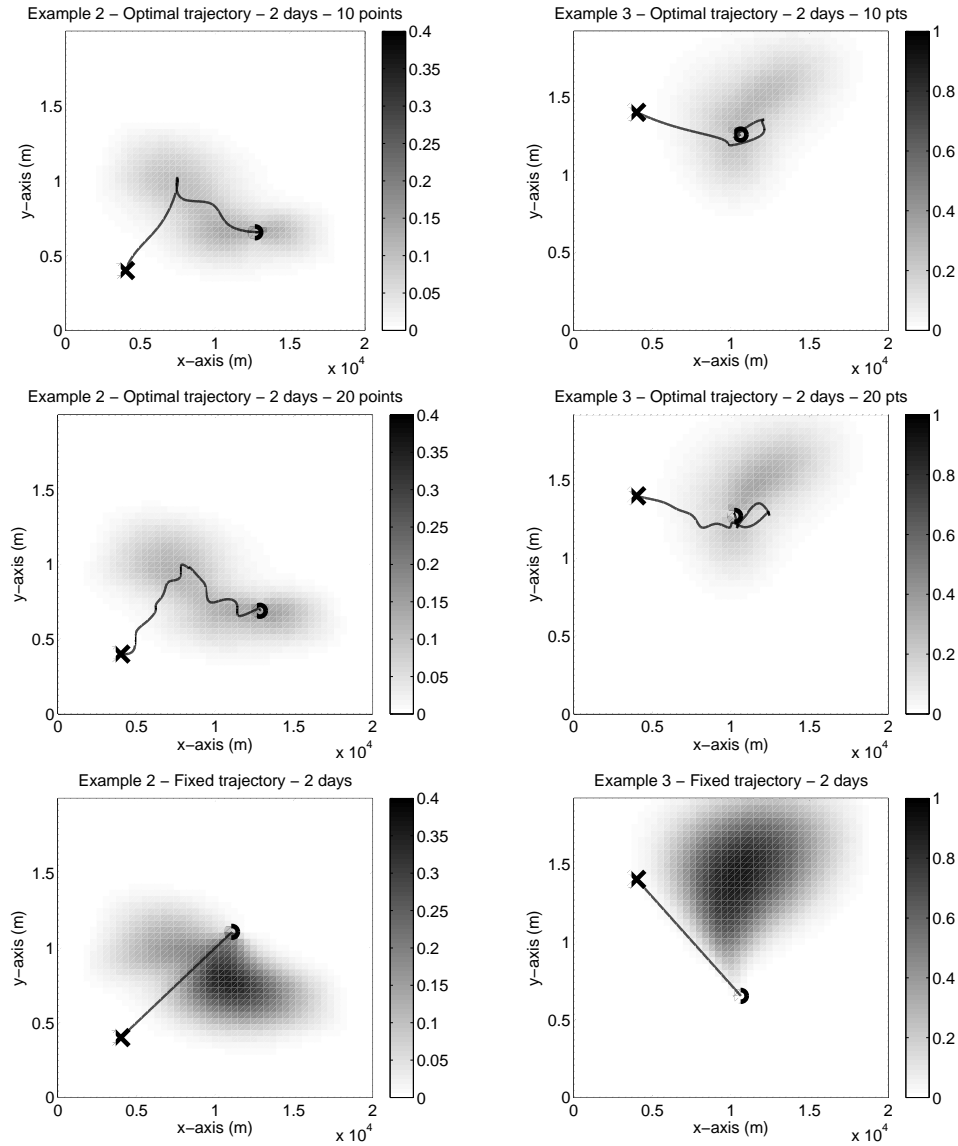


Figure 6.11: Final concentration considering the optimal trajectories, obtained by considering 10 (Top) and 20 (Center) interpolation points, and fixed trajectory (Bottom) for examples (Left) 2 and (Right) 3. The initial position (**X**), the final position (**o**) and the trajectory (–) of the pump are also shown. Finally, the prefixed trajectories are presented in the right figures in gray.

an optimization problem designed to maximize the amount of pumped oil during a fixed time interval. We have used the GMSA, presented in Chapter 2, to solve our optimal trajectory problem. This approach has been validated by considering three numerical examples, based on real oil spots shapes. The results show the efficiency of our method. The developed tool can then be used for real cases.

The next issue to be investigated is the possibility of combining explicit and implicit schemes for speeding-up the computational time. We will investigate splitting and un-split schemes and this might lead us to use other upwinding scheme like superbee [168] to discretized the advection terms in the model. Furthermore, we will need to control the propagation of the diffusion (avoiding infinite speed and artificial diffusion) by replacing the linear diffusion term in (6.2.3) by a nonlinear one.

Chapter 7

A Variance-Expected Compliance Model for the Optimization of Structure Submitted to Random Loads

Abstract - *In this Chapter, we introduce a model to find robust structures for a given main load and its perturbations. In Section 7.1, we show the mathematical formulation of an original variance-expected compliance model used for the optimization of truss structures. In particular, we study the interest of this model on two 3-D benchmark test cases and compare the obtained results with those given by an expected compliance model. Then in Section 7.2, we adapt this methodology to the case of topology optimization. In order to illustrate the interest of our approach, we consider a 2D and 3D benchmark experiments and compare the model outputs with the ones given by a single load approach.*

7.1 Optimization of truss structures submitted to random loads

7.1.1 Introduction

Trusses are mechanical structures consisting of an ensemble of slender bars, connecting some pairs of nodal points. The bars are supposed to be made of a linearly elastic, isotropic and homogeneous material. Long bars overlapping small ones are not allowed. They are designed to support some external nodal loads taking into account certain mechanical properties of the bar material.

Using the *ground structure approach* [55], we focus on the case where the goal is to find the bar volumes which minimize the *compliance* (a measure of the structure energy) of the truss under mechanical equilibrium and total volume constraints.

Optimal trusses can be unstable under load perturbations (see for example [19]). Therefore, in order to design robust truss structures, it is necessary to include these perturbations into the model. In this context, several approaches may be considered. For instance, the *multiload model* which, in its standard form, consists of minimizing a weighted average of compliances associated with a finite set of loads [3]. In the *worst case*

design, the objective is to minimize the maximum compliance under a finite number of loads [1, 2]. Finally, in the same direction, the *ellipsoid method* considers a continuum of loads defined by a particular ellipsoid [19].

Assuming that the loads vary randomly, Alvarez and Carrasco [5] propose the problem of finding the truss of minimum *expected compliance*. They show that this problem can be reformulated as a multiload-type problem, but with a different interpretation of the set of possible loads. Since multiload problems can be equivalently formulated as a convex finite-dimensional minimization problem, the expected-compliance model may be solved by considering, for instance, interior point methods (see [3, 22, 21, 142]).

The stochastic setting given in [5] allows us to consider the *minimum variance-expected compliance problem*, where the variance of the compliance is included into the model. This stochastic model is equivalent to a nonlinear programming problem. However, the variance formulation introduces a non-convex term, that makes this type of problems harder to solve numerically than the minimum expected compliance model.

This Section focuses on this variance-expected compliance model. Considering this formulation and using the SDMSA-L2, presented in Section 2.3.1, we design particular optimal trusses. Then, we study their mechanical stability and compare the results with those obtained with classical formulations.

In Section 7.1.2, we recall the well-known *minimum compliance truss model* and the *multiload model* [19, 1]. In Section 7.1.3, we present the mathematical formulation of the *variance-expected compliance model*. Finally, in Section 7.1.4, those models are analyzed and compared, in terms of robustness, considering two particular 3-D benchmark test cases.

7.1.2 Standard Minimum Compliance Truss Design

Set $n = d \cdot N - s$ to be the number of *degrees of freedom* of a considered structure consisting of $N \geq 2$ nodes, where $s \geq 0$ is the number of fixed nodal coordinate directions (i.e., coordinates corresponding to support conditions are removed) and either $d = 2$ for planar trusses or $d = 3$ for three-dimensional ones. Therefore, the nodal displacements can be described entirely by the n *global reduced coordinates* of the structure. Let $m \geq n$ be the number of potential bars in the truss structure. Long bars overlapping small bars are not allowed. Thus, $m \leq N(N - 1)/2$. We denote by $\lambda_i \geq 0$ the normalized volume of the i -th bar, with $i \in \{1, \dots, m\}$, and $\lambda := (\lambda_1, \dots, \lambda_m)$. External loads are applied only at nodal points and are described, in global reduced coordinates, by a vector $f \in \mathbb{R}^n$. For the sake of simplicity, we assume that each bar is subjected only to axial tension or compression (i.e., we neglect large deflections, bending effects and gravity). In this case, the mechanical response of the truss is described by the *elastic equilibrium equation* (see [1])

$$K(\lambda)u_\lambda = f, \quad (7.1.1)$$

where $u_\lambda \in \mathbb{R}^n$ is the nodal displacements vector in global reduced coordinates and $K(\lambda) \in \mathbb{R}^{n \times n}$ is the *stiffness matrix* of the truss.

Matrix $K(\lambda)$ can be written as (see [55])

$$K(\lambda) = \sum_{i=1}^m \lambda_i K_i, \quad (7.1.2)$$

where $\lambda_i \geq 0$ is the volume of the i -th bar and $K_i \in \mathbb{R}^{n \times n}$ is a dyadic¹ matrix which corresponds to the specific stiffness matrix of the i -th bar in global reduced coordinates.

Given $\lambda \in \mathbb{R}^m$, the compliance value of the corresponding structure is $\frac{1}{2} f^T u_\lambda$. Therefore, the problem of finding the minimum *compliance* truss for a normalized volume constraint of material is given by (see [3, 20])

$$\min_{\lambda \in \Delta_m} \left\{ \frac{1}{2} f^T u_\lambda \right\}, \quad (7.1.3)$$

where u_λ satisfies $K(\lambda)u_\lambda = f$ and $\Delta_m := \{\lambda \in \mathbb{R}^m \mid \lambda \geq 0, \sum_{i=1}^m \lambda_i = 1\}$. This problem is known as the *single load model*.

Remark 21 *It is easy to see that if $\lambda \in \Delta_m$ and $u_\lambda, v_\lambda \in \mathbb{R}^n$ satisfy (7.1.1), then $f^T u_\lambda = f^T v_\lambda$. Thus, the value of the objective function in (7.1.3) does not depend on the choice of the equilibrium displacement vector $u_\lambda \in \mathbb{R}^n$ satisfying (7.1.1), and Problem (7.1.3) can be viewed as an optimization problem in which u_λ is an auxiliary variable. Additionally, if we assume that $K(\lambda)$ is definite positive for some $\lambda \in \Delta_m$, then the existence of a minimum value in (7.1.3) is ensured [5, 20]. We point out that this assumption does not allow rigid body motions (i.e., movements of the structure without changing the relative position among the nodes) or mechanisms (i.e., combination of resistant bodies interconnected in such a way that the application of f causes some of these bodies move). See [20, 72] for more details.*

Remark 22 *For the sake of simplicity, the self-weight of the bars is not included in the formulation of the model. If we want to consider the self-weight, we have to replace f by $f + g(\lambda)$ in Problem (7.1.3), where $g(\lambda) = \sum_{i=1}^m \lambda_i g_i$, with g_i being a specific nodal gravitation force vector (see [20]).*

Problem (7.1.3) is equivalent to the following min-max problem

$$\min_{\lambda \in \Delta_m} \max_{x \in \mathbb{R}^n} \left\{ f^T x - \frac{1}{2} x^T K(\lambda) x \right\}. \quad (7.1.4)$$

We note that u_λ satisfying (7.1.1) is a solution (not necessarily unique) of the maximization problem in (7.1.4).

Using a min-max theorem presented in, e.g., [238] - §37, we can rewrite (7.1.4) as

$$- \min_{x \in \mathbb{R}^n} \max_{\lambda \in \Delta_m} \left\{ \frac{1}{2} x^T K(\lambda) x - f^T x \right\}. \quad (7.1.5)$$

¹A matrix $A \in \mathbb{R}^{n \times n}$ is called dyadic when $A := bb^T$ for some $b \in \mathbb{R}^n$.

Then, using the fact that $K(\lambda)$ depends linearly on λ , and therefore, the maximum value is achieved at some vertices of the polytope Δ_m , we obtain that Problem (7.1.5) is equivalent to

$$-\min_{x \in \mathbb{R}^n} \max_{1 \leq i \leq m} \left\{ \frac{1}{2} x^T K_i x - f^T x \right\}. \quad (7.1.6)$$

According to [3, 1], the previous problem can be rewritten as the following convex quadratic minimization problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n, \mu \in \mathbb{R}} \quad & \left\{ \frac{1}{2} \mu^2 - f^T x \right\}, \\ \text{such that:} \quad & x^T K_i x \leq \mu^2, \text{ for } i = 1, \dots, m. \end{aligned} \quad (7.1.7)$$

A solution of (7.1.3) can be obtained by computing the multipliers of the Karush-Kuhn-Tucker (KKT) conditions of Problem (7.1.7) (see [3] for more details).

Taking into account that matrices K_i are dyadic (i.e., $K_i = b_i b_i^T$, for some $b_i \in \mathbb{R}^n$), it is possible to show (see [3, 1]) that Problem (7.1.7) is equivalent to the following linear programming problem

$$\begin{aligned} \max_{x \in \mathbb{R}^n} \quad & \left\{ f^T x \right\}, \\ \text{such that:} \quad & -1 \leq b_i^T x \leq 1, \text{ for } i = 1, \dots, m, \end{aligned} \quad (7.1.8)$$

and therefore, might be solved by using a standard primal-dual LP method, such as Mehrotra's predictor-corrector method (see, e.g., [192, 279]).

Nevertheless, numerical results using this model show that the optimal solutions may be unstable with respect to the mechanical equilibrium, even under small perturbations in the principal load [19, 1]. In fact, there are several examples showing some optimal structures that, under small perturbations, give infinite compliance.

In order to handle this inconvenience, we may consider the *multiload model* (see [3, 22]) given by

$$\min_{\lambda \in \Delta_m} \left\{ \frac{1}{2} \sum_{j=0}^k \gamma_j (f_j)^T u_{\lambda,j} \right\} \quad (7.1.9)$$

where $u_{\lambda,j}$ satisfies $K(\lambda) u_{\lambda,j} = f_j$, for $j = 0, \dots, k$. In this formulation, f_0 can be interpreted as the main load, $(f_j)_{j=1}^k$ are k possible perturbations of f_0 and $\gamma_j \geq 0$ corresponds to the influence of f_j .

We point out that Problem (7.1.9) can be expressed as a problem of type (7.1.3), by rewriting it as follows

$$\min_{\lambda \in \Delta_m} \left\{ \frac{1}{2} \hat{f}^T U_\lambda \right\}, \quad (7.1.10)$$

where U_λ satisfies $\hat{K}(\lambda) U_\lambda = \hat{f}$, $\hat{f}^T := (\gamma_0 f_0^T, \dots, \gamma_k f_k^T)$, $\hat{K}(\lambda) := \text{bdiag}_{k+1}(K(\lambda))$ and $\text{bdiag}_k(A)$ denotes the block diagonal matrix built by considering k times matrix A . Thus, as for Problem (7.1.3), Problem (7.1.9) is equivalent to

$$\begin{aligned} \min_{x \in \mathbb{R}^{(k+1) \times n}, \mu \in \mathbb{R}} \quad & \left\{ \frac{1}{2} \mu^2 - \hat{f}^T x \right\}, \\ \text{such that:} \quad & x^T \hat{K}_i x \leq \mu^2 \text{ for } i = 1, \dots, m. \end{aligned} \quad (7.1.11)$$

where $\hat{K}_i := \text{bdiag}_{k+1}(K_i)$. Some numerical algorithms, based on interior point methods, have been developed to solve this kind of problem (see, for instance, [21, 142]). As previously, using the KKT conditions of Problem (7.1.11), we can obtain a solution of (7.1.9) (see [1]). We point out that in this case, since matrices \hat{K}_i are not dyadic, Problem (7.1.9) cannot be transformed into a linear problem.

However, in order to model a stable structure submitted to randomly perturbed loads using the *multiload model*, we should take into account all the possible perturbations $(f_j)_{j=1}^k$, (or at least, numerically, a high enough number of them), see, e.g., [1]. Thus, depending of the value of k , the computational complexity of this approach can be high. In order to handle this difficulty, it is possible, in some cases, to consider another method based on a good representation of $(f_j)_{j=1}^k$ and depending on the variance of f_j , $j = 0, \dots, k$, with respect to f_0 (see [5]). This methodology, is briefly presented in Section 7.1.3.

7.1.3 Variance-Expected Compliance Model

General Formulation

According to the definitions and notations introduced in Section 7.1.2, we define the function $\Psi: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$ as

$$\Psi(\xi, \lambda) := \begin{cases} \frac{1}{2}(f + \xi)^T u_\lambda, & \text{if } \lambda \in \Delta_m \text{ and } \exists u_\lambda \in \mathbb{R}^n \\ & \text{such that } K(\lambda)u_\lambda = f + \xi, \\ +\infty, & \text{otherwise.} \end{cases} \quad (7.1.12)$$

Function Ψ is proper², lower semi-continuous and convex (see [5]). Therefore, for each $\lambda \in \Delta_m$, the function

$$\Psi(\cdot, \lambda): (\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n)) \rightarrow (\mathbb{R} \cup \{+\infty\}, \bar{\mathcal{B}}(\mathbb{R}))$$

is measurable. Here, $\mathcal{B}(\mathbb{R}^n)$ and $\bar{\mathcal{B}}(\mathbb{R})$ stand for the Borel σ -algebra of \mathbb{R}^n and $\mathbb{R} \cup \{+\infty\}$, respectively.

Next, we assume that ξ is a random variable corresponding to an uncertain perturbation of f . More precisely, let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space and consider a measurable function

$$\begin{aligned} \xi: (\Omega, \mathcal{A}) &\rightarrow (\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n)) \\ \omega &\mapsto \xi(\omega). \end{aligned}$$

According to this setting, we study the following stochastic minimization problem:

$$\min_{\lambda \in \Delta_m} \{\alpha \mathbb{E}[\Psi(\xi, \lambda)] + \beta \text{Var}[\Psi(\xi, \lambda)]\}, \quad (7.1.13)$$

where Ψ is defined by (7.1.12), $\alpha, \beta \geq 0$, $\alpha + \beta > 0$, $\mathbb{E}(\cdot)$ denotes the expected value of the corresponding random function and $\text{Var}(\cdot)$ denotes its variance.

²A convex function is called proper, if and only if its epigraph is not empty and does not contain vertical lines (see [238]).

Remark 23 Let $\lambda \in \Delta_m$ and consider $V_\lambda := \{K(\lambda)v \mid v \in \mathbb{R}^n\}$. If the probability $\mathbb{P}\{f + \xi \notin V_\lambda\} > 0$, then the value of $\alpha \mathbb{E}[\Psi(\xi, \lambda)] + \beta \text{Var}[\Psi(\xi, \lambda)] = +\infty$. In this case, the structure associated to the volume λ cannot sustain the perturbed load and λ is considered as a non-admissible point of Problem (7.1.13).

We point out that, if $\alpha = 1$, $\beta = 0$ and ξ is a random variable with finite support ξ_1, \dots, ξ_k , then, by denoting $\gamma_j := \mathbb{P}\{\xi = \xi_j\}$ and $f_j := f + \xi_j$, for $j = 1, \dots, k$, the model described in (7.1.13) is the same as the one given by (7.1.9). Therefore, model (7.1.13) extends the well-known *multiload model* (7.1.9).

General Results

The first result provides an explicit expression for (7.1.13), with $\alpha = 1$ and $\beta = 0$, when the perturbation ξ is a continuous random variable. In the sequel, given a square matrix $A = (a_{ij})$, we denote by $\text{Tr}(A)$ the trace of A (i.e., $\text{Tr}(A) := \sum_i a_{ii}$).

Theorem 24 Let $\xi \in \mathbb{R}^n$ be a continuous random vector, generated in a subspace of dimension $k \in \mathbb{N}$, such that $\mathbb{E}(\xi) = 0$ and its covariance matrix is given by $\text{Cov}(\xi) := PP^T$, with $P \in \mathbb{R}^{n \times k}$ (such a decomposition always exists [245]). In that case, $\mathbb{E}[\Psi(\xi, \lambda)] = \frac{1}{2}f^T u_\lambda + \frac{1}{2} \text{Tr}(P^T U_\lambda)$, with u_λ and U_λ satisfying $K(\lambda)u_\lambda = f$ and $K(\lambda)U_\lambda = P$, respectively. Thus, the corresponding minimum expected compliance design problem (7.1.13), with $\alpha = 1$ and $\beta = 0$, can be rewritten as

$$\min_{\lambda \in \Delta_m} \left\{ \frac{1}{2}f^T u_\lambda + \frac{1}{2} \text{Tr}(P^T U_\lambda) \right\}, \quad (7.1.14)$$

where u_λ satisfies $K(\lambda)u_\lambda = f$, and U_λ satisfies $K(\lambda)U_\lambda = P$.

The proof of this theorem can be found in [5].

As in Problem (7.1.3), the value of the objective function in (7.1.14) is independent of $u_\lambda \in \mathbb{R}^n$ and $U_\lambda \in \mathbb{R}^{n \times k}$ satisfying $K(\lambda)u_\lambda = f$ and $K(\lambda)U_\lambda = P$, respectively. Therefore, Problem (7.1.14) can be seen as a minimization problem with λ as the only unknown (design variable).

We point out that, denoting by $f_0 = f$ and f_j the j th-column of matrix P , for $j = 1, \dots, k$, Equation (7.1.14) may be rewritten as a multi-load-type problem with $k + 1$ loads (see Problem (7.1.9))

$$\min_{\lambda \in \Delta_m} \left\{ \frac{1}{2} \sum_{j=0}^k (f_j)^T u_{\lambda,j} \right\}. \quad (7.1.15)$$

where $u_{\lambda,j}$ satisfies $K(\lambda)u_{\lambda,j} = f_j$, for $j = 0, \dots, k$

In this case, f_j can be interpreted as a load perturbation of the main force f_0 (see [5]). Thus, in order to construct robust structures in this continuous model, it is not necessary to consider explicitly all the loading possibilities but a good representation of them, according to the covariance matrix $P^T P$.

Now, we deal with the case $\beta \neq 0$ in the formulation of (7.1.13) and we rewrite that stochastic model as a minimization problem.

Theorem 25 *Let $\xi \in \mathbb{R}^n$ be a continuous random vector, generated in a subspace of dimension $k \in \mathbb{N}$, such that the distribution of ξ is a n -multivariate normal, $\mathbb{E}(\xi) = 0$ and its covariance matrix is given by $\text{Cov}(\xi) := PP^T$, where $P \in \mathbb{R}^{n \times k}$ (i.e., $\xi \sim \mathcal{N}_n(0, PP^T)$).*

In that case, $\mathbb{E}[\Psi(\xi, \lambda)] = \frac{1}{2}f^T u_\lambda + \frac{1}{2}\text{Tr}(P^T U_\lambda)$ and $\text{Var}[\Psi(\xi, \lambda)] = \frac{1}{2}\text{Tr}(P^T U_\lambda)^2 + f^T U_\lambda U_\lambda^T f$, with u_λ and U_λ satisfying $K(\lambda)u_\lambda = f$ and U_λ satisfies $K(\lambda)U_\lambda = P$, respectively. Thus, Problem (7.1.13) can be rewritten as

$$\min_{\lambda \in \Delta_m} \left\{ \alpha \left(\frac{1}{2}f^T u_\lambda + \frac{1}{2}\text{Tr}(P^T U_\lambda) \right) + \beta \left(\frac{1}{2}\text{Tr}(P^T U_\lambda)^2 + f^T U_\lambda U_\lambda^T f \right) \right\}, \quad (7.1.16)$$

where u_λ satisfies $K(\lambda)u_\lambda = f$, and U_λ satisfies $K(\lambda)U_\lambda = P$.

The proof of this theorem can be found in [5].

Remark 26 *Theorems 24 and 25 are useful because, when their assumptions are satisfied, they give an explicit expression of Problem (7.1.13). In general cases, it could be difficult to evaluate $\mathbb{E}[\Psi(\xi, \lambda)]$ and $\text{Var}[\Psi(\xi, \lambda)]$. We can consider, for instance, a Monte-Carlo algorithm (see [136]) to approximate those values. However, this method, and thus the resolution of (7.1.13), is numerically expensive.*

Remark 27 *In Theorem 25, the random vector ξ can be written as $\xi = \sum_{i=1}^k \xi_i V_i$, where $\xi_i \sim \mathcal{N}(0, \sigma_i^2)$ and $\{V_i\}_{i=1, \dots, k}$ is a family of linear independent vectors in \mathbb{R}^n . Thus, in global coordinates, P is of the form $P := [P_1, P_2, \dots, P_k]$ with $P_i := \sigma_i V_i$ (see [245] for more details).*

Remark 28 *It is worth pointing out the nonlinearity of this problem. Thus, it would be interesting to develop a primal-dual formulation of (7.1.13), which allows us to implement standard numerical resolution methods for this kind of problems as those used to solve Problem (7.1.9). Nevertheless, the computational time and the number of design variables (i.e., volumes of the bars) can be reduced, in some cases, by exploiting the symmetry of the problem. For instance, we can assign the same volume value to all bars included in the same orbit (here, an orbit of a bar δ is a set of bars to which δ can be moved by some element of Θ , where Θ is a group acting on the bars, such as rotations or symmetries in \mathbb{R}^3), following the ideas of [146] and [17]. In Section 7.1.4, we will apply this technique to the considered optimization problems in order to reduce their computational complexity.*

In next Section, we present various numerical experiments used to check the advantages of the formulation (7.1.13).

7.1.4 Numerical Study

Numerical Problem

In order to perform a numerical study to see the interest of Problem (7.1.13) and the choice of (α, β) , we consider two 3-D benchmark structures (i.e., $d = 3$):

- The first one, denoted by **Michel**, was presented in [142]. It is composed of a set of $N = 27$ nodes of coordinates (i, j, k) , with $i, j, k \in \{0, 0.5, 1\}$. All these nodes can be connected by bars (i.e., $m = 302$). The 9 nodes of coordinates $(0, j, k)$, with $j, k \in \{0, 0.5, 1\}$, are fixed (i.e., $s=27$). A main vertical load $f \in \mathbb{R}^3$ (expressed, in this section, in 3D *local coordinates*) is applied to the node with coordinates $(1, 0.5, 0)$.
- The second one, denoted by **Dome**, was proposed in [216]. It is a dome composed of a set of $N = 101$ nodes: one node denoted by γ_1 , of coordinates $(0, 0, 0.16432)$, a set of 50 nodes, denoted by Γ_2 , of coordinates $(\frac{1}{2} \cos(2i\pi/50), \frac{1}{2} \sin(2i\pi/50), 0.12432)$, with $i \in \{1, 2, \dots, 50\}$, and a set of 50 fixed nodes, denoted by Γ_3 , of coordinates $(\cos((2i+1)\pi/50), \sin((2i+1)\pi/50), 0)$, with $i \in \{1, 2, \dots, 50\}$. The node of index i in Γ_2 can be connected by a bar to γ_1 , to the nodes of index $i-1$ and $i+1$ in Γ_2 and to the nodes of index i and $i+1$ in Γ_3 . Thus, the number of bars is $m = 200$. A main vertical load $f \in \mathbb{R}^3$ (expressed in 3D local coordinates) is applied to the node γ_1 .

These structures are depicted in Figure 7.1.

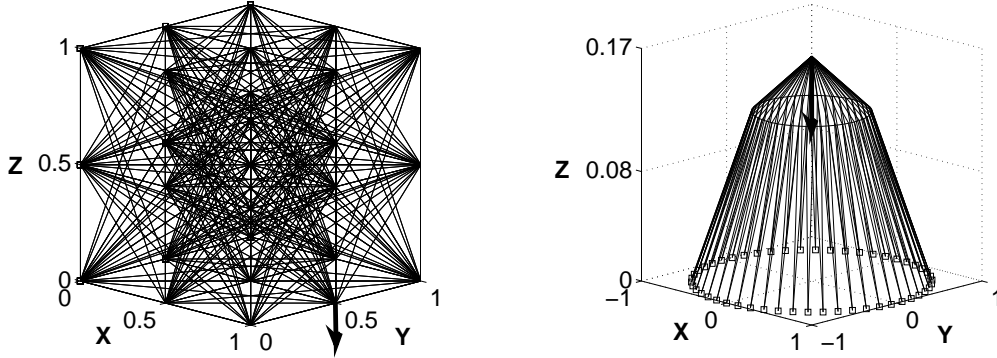


Figure 7.1: 3-D benchmark structures **Michel** and **Dome** presented in Section 7.1.4. Possible bars are represented by lines (—), fixed nodes are represented by squares (\square) and the node, where the main load f is applied, is represented by an arrow (\downarrow).

Problem (7.1.3) can be solved, for these benchmark structures, by using the dual approach presented in [3, 1, 41]. The obtained solutions, presented in Figure 7.2, are mechanically stable to their main load f but are unstable in cases of small perturbations of this load (they generate high compliance values).

Thus, since we are interested in generating structures that are stable to small perturbations of the design force f , we consider a random load ξ of law $\mathcal{N}_n(0, PP^T)$ applied to the same node as f and situated in the plane orthogonal to f . The idea is to illustrate the influence of this kind of perturbations on the main load in the final structure. We point out that there is no limitation in the model with respect to the number of extra load forces or directions that can be considered, and, of course, we can also add perturbations in the same direction of f . Nevertheless, in order to obtain robust structural designs, it

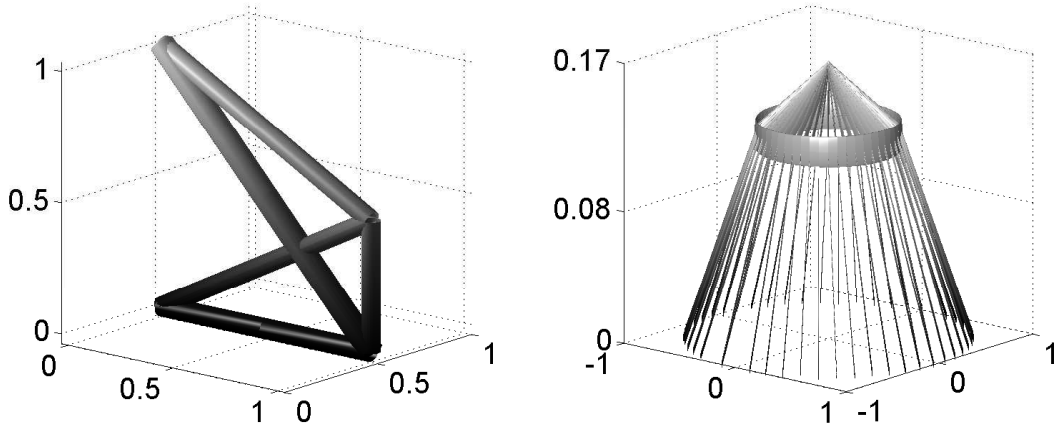


Figure 7.2: Solutions obtained with the dual approach described in [3, 1, 41] considering the 3-D benchmark structures **Michel** and **Dome** presented in Section 7.1.4.

should be enough to take f as an upper bound of the main force and to consider only perturbations in the plane orthogonal to f .

We solve the corresponding Problem (7.1.13), considering the formulation given by Theorem 25, with various sets of values of (α, β) . More precisely, (α, β) is chosen in a set $\Sigma := \{(\alpha_i, \beta_i)\}_{i=1}^S$ with $(\alpha_i, \beta_i) \in [0, 1] \times [0, 1]$ and $S \in \mathbb{N}$.

In order to solve (7.1.16) for the **Michel** and **Dome** structures, which seem to be non-convex problems with several local minima (see Figure 7.3), we use the SDMSA-L2, presented in Section 2.3.1 (with $A_2, t_0 = 50, t_{l_1} = 5, t_{l_2} = 5$ and $\epsilon = 0$).

Furthermore, as observed in Remark 28, we can exploit the symmetry of those two optimization problems to reduce the number of design variables:

- In the case of the **Michel** structure, we have symmetry with respect to the plane $\{(x, y, z) \in \mathbb{R}^3 / y = 0.5\}$ that allows to reduce the problem to 159 variables instead of 302 (we consider the bars in the subspace defined by $\{(x, y, z) \in \mathbb{R}^3 / y \leq 0.5\}$ and we can obtain the other bars by applying this symmetry).
- In the case of the **Dome** structure, we have rotational symmetry of angle $\frac{\pi}{50}$ with respect to the axis $\{(x, y, z) \in \mathbb{R}^3 / x = 0, y = 0\}$ and symmetries with respect to the vertical planes containing the bars starting from γ_1 . In this case, we can reduce the number of problem variables to 3 instead of 200 (one bar in each orbit, i.e., one bar connecting γ_1 to Γ_2 , one bar connecting two consecutive points in Γ_2 and one bar connecting a point of Γ_2 to the closer point in Γ_3).

The obtained solutions are denoted by $\{\lambda_{(\alpha, \beta)}^M\}_{(\alpha, \beta) \in \Sigma}$ for the **Michel** structure and $\{\lambda_{(\alpha, \beta)}^D\}_{(\alpha, \beta) \in \Sigma}$ for the **Dome** structure. It is interesting to notice that, for $(\alpha, \beta) = (1, 0)$, the solutions $\lambda_{(1, 0)}^M$ and $\lambda_{(1, 0)}^D$ are the same solutions as the ones obtained by using the dual approach described in [1]. In order to have a qualitative comparison of these structures

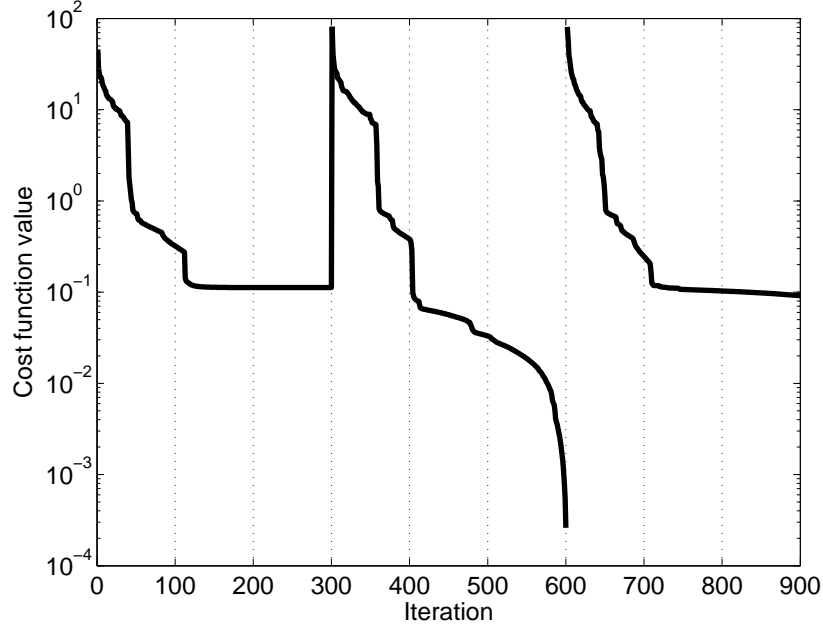


Figure 7.3: Part of the convergence history obtained from the SDMSA-L2 used to solve Problem (7.1.13) with $(\alpha, \beta) = (1, 0.25)$ and considering the **Michel** structure. As we can observe, the steepest descent algorithm visits various local attraction basins (i.e., the cost function seems to have several local minima).

$\{\lambda_{(\alpha, \beta)}^M, \lambda_{(\alpha, \beta)}^D\}_{(\alpha, \beta) \in \Sigma}$, we analyze their robustness when they are submitted to random loads, their shape and their compliance iso-contours. More precisely:

- To study the robustness of a particular structure λ , we consider the random variable $\Phi_\lambda := \Psi(\xi, \lambda)$, where Ψ is defined by (7.1.12), and we approximate its density function ρ_{Φ_λ} using a *Monte-Carlo* approach [136] that generates $M \in \mathbb{N}$ possible values of ξ . Then, we compute the maximum value of ρ_{Φ_λ} and a particular risk measure of Φ_λ . *Risk measures*, on $L^\infty(\Omega, \mathcal{A}, \mathbb{P})$, are mappings $\varpi : L^\infty(\Omega, \mathcal{A}, \mathbb{P}) \rightarrow \mathbb{R}$ where $(\Omega, \mathcal{A}, \mathbb{P})$ is a probability space (a complete presentation can be found in [12, 11]). They are used in various areas, such as financial analysis [136], in order to study the worst cases (here, the random loads generating the highest structure compliances). Here, we focus on a particular and popular risk measure called the *Coherent-Value at Risk* (C-VaR) [12], defined as:

$$\text{C-VaR}_\nu(\chi) := \frac{1}{\nu} \int_0^\nu \inf \left\{ z \in \mathbb{R} \text{ s.t. } \int_0^z 100\rho_\chi(x)dx > (100 - y) \right\} dy,$$

where ν is a percentile, $\chi \in L^\infty(\Omega, \mathcal{A}, \mathbb{P})$ and ρ_χ is the density function of χ . C0-VaR_ν corresponds to the average value of the ν % worst cases of χ (i.e., the ν % highest values of χ). In our case, we take $\chi = \Phi_\lambda$.

α	1	1	1	1	1	1	0.75	0.5	0.25	0
β	0	0.1	0.25	0.5	0.75	1	1	1	1	1
Michel										
EC	32	33	34	36	38	40	42	48	56	10^7
Vari	47	31	24	20	17	15	13	10	8	2
Max	120	114	95	91	87	88	90	92	104	10^7
C-VaR₁	60	55	53	54	55	56	58	62	70	10^7
Dome										
EC	479	536	590	653	699	737	779	845	971	1394
Vari	913	282	203	162	143	132	124	114	103	94
Max	874	757	809	830	865	927	921	980	1105	1529
C-VaR₁	618	613	656	711	754	790	830	894	1018	1439

Table 7.1: Summary of the results obtained by solving Problem (7.1.13) considering structures **Michel** and **Dome** and the set of coefficients $(\alpha, \beta) \in \Sigma$: Expected compliance (**EC**), Variance of the compliance (**Vari**), maximum compliance value (**Max**) and Coherent Value at Risk of the compliance (**C-VaR₁**) of the solutions.

- To study the compliance iso-contours of a structure λ in function of the loads, we consider $J_\lambda : \mathbb{R}^3 \rightarrow \mathbb{R}$ defined by:

$$J_\lambda(x_1, x_2, x_3) := \frac{1}{2} \hat{f}^T(x_1, x_2, x_3)u,$$

where $\hat{f}(x_1, x_2, x_3) := x_1 \frac{f}{\|f\|} + x_2 d_1 + x_3 d_2$, d_1 and d_2 are two orthogonal loads applied at the same node as f and situated in the plane orthogonal to f , and $u \in \mathbb{R}^n$ is a solution of $K(\lambda)u = \hat{f}$. Then, fixing $r \in \mathbb{R}$, we compute numerically one iso-contour of this function, considering the values of $x \in \mathbb{R}^3$ such that $J_\lambda(x) = r$.

Numerical Results

We set $f = (0, 0, -100)$, $\xi := \xi_1 V_1 + \xi_2 V_2$ where $\xi_1 \sim \mathcal{N}(0, 625)$, $\xi_2 \sim \mathcal{N}(0, 625)$, $V_1 = (1, 0, 0)$, $V_2 = (0, 1, 0)$ (here, f , V_1 and V_2 are expressed in local coordinates), $\Sigma = \left\{ (1, 0), (1, \frac{1}{10}), (1, \frac{1}{4}), (1, \frac{1}{2}), (1, \frac{3}{4}), (1, 1), (\frac{3}{4}, 1), (\frac{1}{2}, 1), (\frac{1}{4}, 1), (0, 1) \right\}$, $M = 10^6$, $r = 0.1$ (we use this small number, in comparison to the compliance values reported in Table 7.1, in order to be able to draw the iso-contours on the corresponding figures) and $\nu = 1$ (this level is often used since it reduces the impact of too extreme cases [136]).

All the results are reported in Table 7.1. The different solutions and their compliance iso-contours in function of the loads are presented in Figures 7.4 and 7.5. A boxplot [258] representation of the densities ρ_{Φ_λ} , with $\lambda \in \{\lambda_{(\alpha, \beta)}^M, \lambda_{(\alpha, \beta)}^D\}_{(\alpha, \beta) \in \Sigma}$, is shown in Figure 7.6.

As we can observe in Table 7.1 and Figure 7.6,

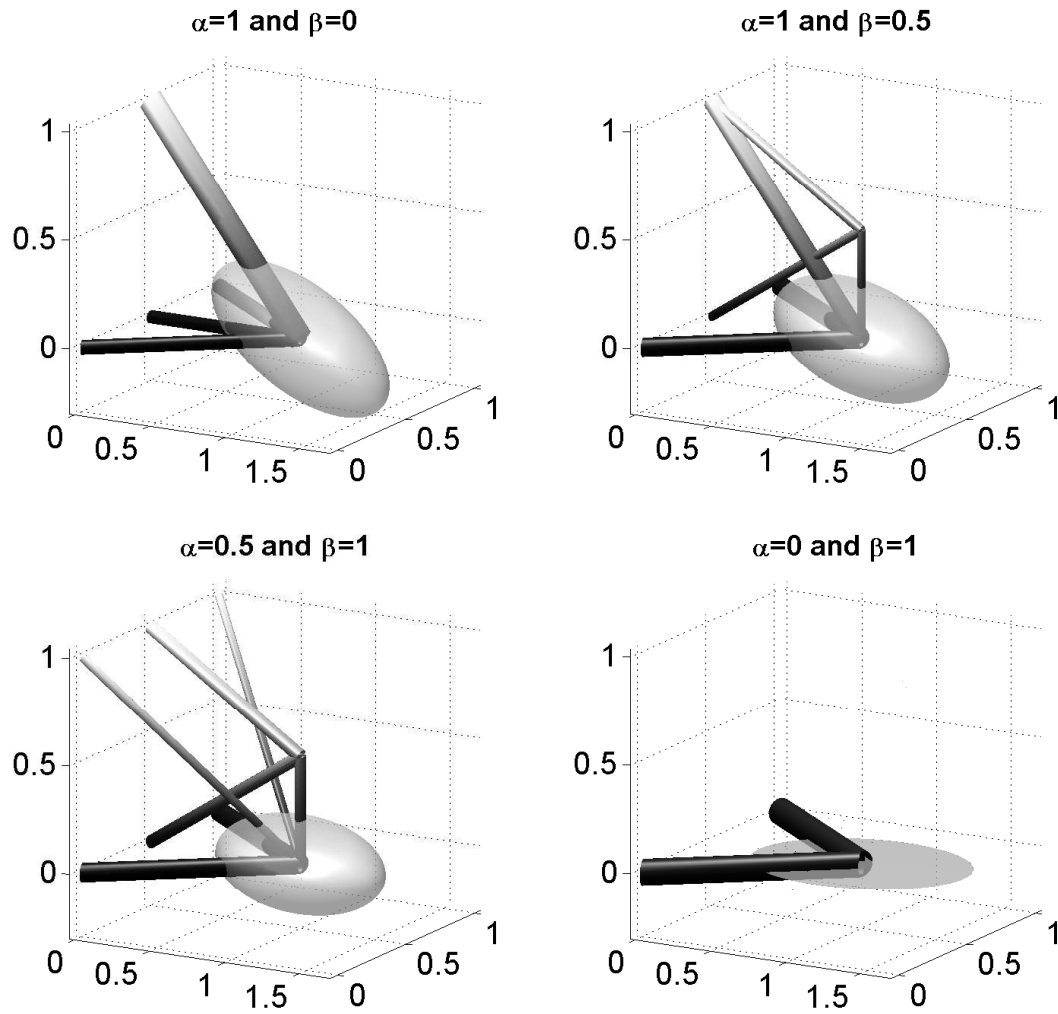


Figure 7.4: Shape and compliance iso-contour (transparent ellipsoid) in function of the loads, of the solutions $\lambda_{\alpha,\beta}^M$ considering (α, β) set to **(Top-Left)** (1,0), **(Top-Right)** (1,0.5), **(Bottom-Left)** (0.5,1) and **(Bottom-Right)** (0,1).

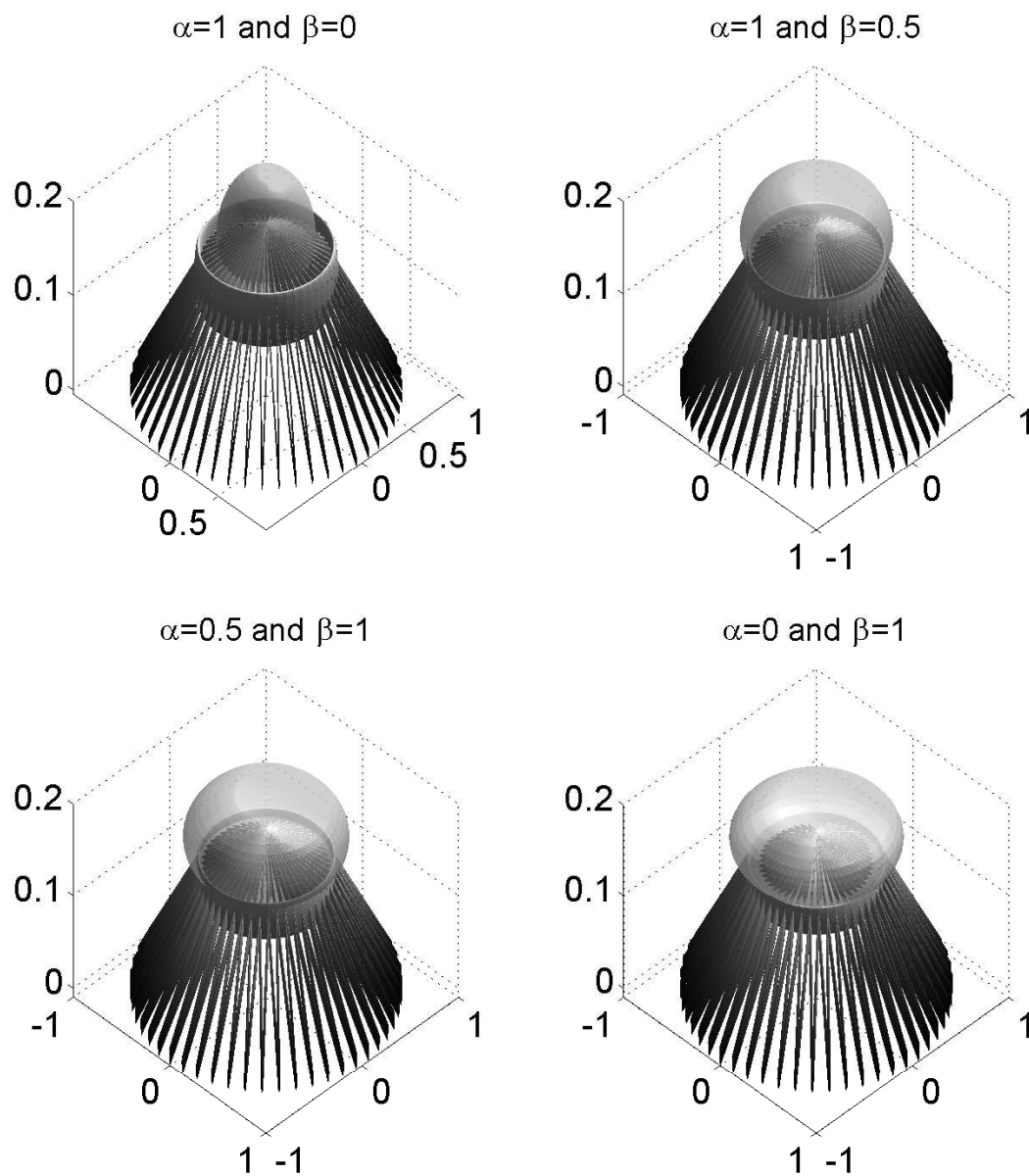


Figure 7.5: Shape and compliance iso-contour (transparent ellipsoid) in function of the loads, of the solutions $\lambda_{\alpha,\beta}^D$ considering (α, β) set to (**Top-Left**) (1,0), (**Top-Right**) (1,0.5), (**Bottom-Left**) (0.5,1) and (**Bottom-Right**) (0,1).

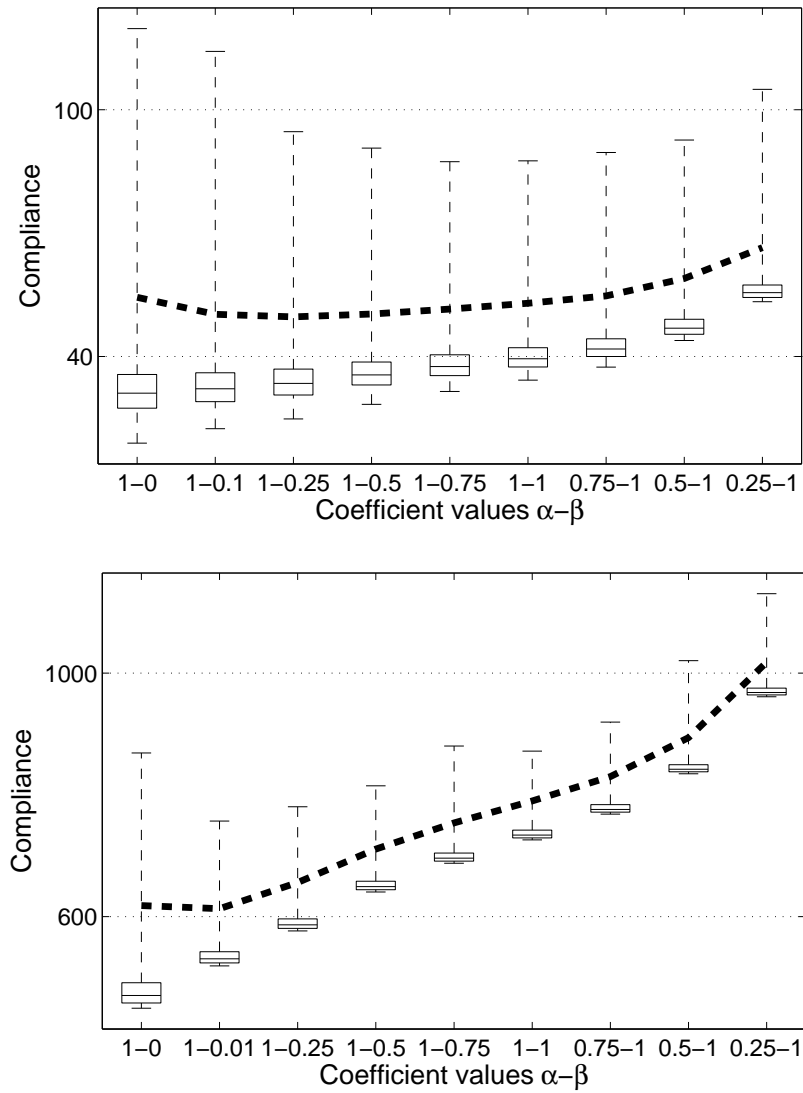


Figure 7.6: Boxplot representation of the density function of the compliance value of the solutions (**Top**) $(\lambda^M_{(\alpha,\beta)})$ and (**Bottom**) $(\lambda^D_{(\alpha,\beta)})$ with $(\alpha, \beta) \in \Sigma \setminus \{(0, 1)\}$. Boxplots are composed by a box, which has lines at the lower quartile, median, and upper quartile values, and whiskers that extend from each end of the box to the lower and higher compliance value. For each solution, the $C\text{-Var}_1$ of the compliance value is represented by the dashed line (- -) intersecting the associated boxplot.

- For the **Michel** structure, the solution $\lambda_{(1,0)}^M$ is less stable to perturbations of the main load, considering the 1% worst cases (C-VaR₁), than the solutions $\lambda_{(1,0.1)}^M$, $\lambda_{(1,0.25)}^M$, $\lambda_{(1,0.5)}^M$, $\lambda_{(1,0.75)}^M$, $\lambda_{(1,1)}^M$ and $\lambda_{(0.75,1)}^M$. This is confirmed by the fact that the maximum compliance value (**Max**) of all considered Monte-Carlo cases is more important for $\lambda_{(1,0)}^M$ than for all other structures, except $\lambda_{(0,1)}^M$.
- For the **Dome** structure, the solution $\lambda_{(1,0.1)}^D$ has the lowest risk measure and the lowest worst cases value. Solutions from $\lambda_{(1,0.25)}^D$ to $\lambda_{(0,1)}^D$ have a higher risk measure than $\lambda_{(1,0)}^D$.

However, in both cases, the value of the expected compliance increases as the value of β increases, becoming less stable to the main load.

Figures 7.4 and 7.5 show that the shape of the structure changes with the evolution of the coefficients α and β . In fact, it seems that there is a mass transfer phenomenon when β goes from 0 to 1:

- In the **Michel** structure, from the upper to the lower bars. This can be observed in Figure 7.4, which shows four different shape configurations, depending on the values of (α, β) . The shape of $\lambda_{(0,1)}^M$ with only two main bars at the bottom, seems to confirm this situation. This is intuitive as those two bars provide a good resistance to horizontal loads and thus, to the perturbations of the main load. In counterpart, they are not stable to f .
- In the **Dome** structure, from the central ring bars to the other bars, especially those starting from Γ_3 . As previously, those bars provide a good structure reinforcement when applying an horizontal perturbation.

As we can also observe in Figures 7.4 and 7.5, the iso-contours are ellipsoids that tend to straighten and flatten in the orthogonal plane of the main load f as β increases. This geometrical evolution corroborates the fact that the structure becomes more robust to the horizontal perturbations of the main load, but more fragile to f .

From all these results, we deduce that considering formulation (7.1.13), with a reasonable balance between α and β , can be helpful to generate more robust structures with respect to perturbations of the main load. For instance,

- Considering the **Michel** structure, $\lambda_{(1,0.25)}^M$ is as a good compromise to $\lambda_{(1,0)}^M$. Indeed, $\lambda_{(1,0.25)}^M$ has the lowest value of C-VaR₁, has the variance value (**Vari.**) reduced by 45 % and the worst case value (**Max**) reduced by 21% in comparison to $\lambda_{(1,0)}^M$ and has an expected compliance value (**E. Cp.**) of 34, close to the best one of 32.
- Considering the **Dome** structure, $\lambda_{(1,0.1)}^D$ should be preferred. It has the lowest risk measure (**C-VaR₁**) and the lowest compliance value of the worst case (**Max**). Its variance value is reduced by 77 % and its expected compliance is increased by 12%, when compared to $\lambda_{(1,0)}^D$.

In Figure 7.7, we show the shapes corresponding to $\lambda_{(1,0.25)}^M$ and $\lambda_{(1,0.1)}^D$.

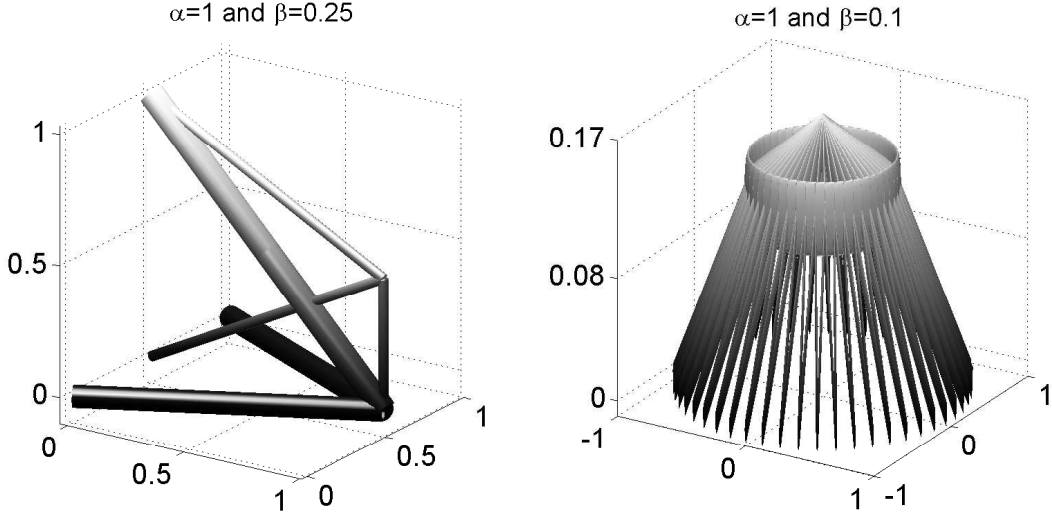


Figure 7.7: Shape of the solutions $\lambda_{(1,0.25)}^M$ (**Left**) and $\lambda_{(1,0.1)}^D$ (**Right**).

7.1.5 Conclusions

We have presented a new variance-expected compliance formulation that has been tested numerically on two 3-D benchmark test cases. This new formulation, combined with an adequate balance between the variance of the compliance and the expected compliance, allows to generate structures that are more stable, in the sense of the worst cases, to perturbations of the main load.

In next Section 7.2, we generalize this formulation to a continuous model (i.e. topology optimization, see [20, 42]) and we carry out new numerical experiments.

7.2 Topology optimization of structure submitted to random loads

7.2.1 Introduction

We consider an elastic homogeneous body $\Omega \subseteq \mathbb{R}^d$, where d is either 2 or 3. We impose that in $\Gamma_u \subseteq \partial\Omega$ (boundary of Ω) the displacements of the body are not allowed and we apply some external load forces f to Ω and g to $\Gamma_t \subseteq \partial\Omega$. A graphical representation of Ω and the external forces is given in Figure 7.8.

In order to simplify the notations, we assume, without loss of generality, that $g = 0$. We note that the results obtained in this paper can be easily extended to the case $g \neq 0$. Furthermore, we consider that our body is composed by a linear material and therefore the displacements can be computed by solving the following system of linear partial differential

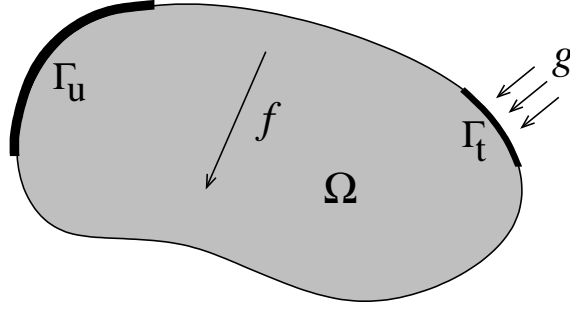


Figure 7.8: Representation of the elastic homogeneous body Ω and the considered external forces f and g .

equations:

$$\begin{cases} -\operatorname{div}(K e(u)) = f, & \text{in } \Omega, \\ u = 0, & \text{on } \Gamma_u, \\ (K e(u)) \cdot n = 0, & \text{on } \partial\Omega \setminus \Gamma_u, \end{cases} \quad (7.2.1)$$

where $f : \Omega \rightarrow \mathbb{R}^d$ corresponds to an external load, $u : \Omega \rightarrow \mathbb{R}^d$ is the vector of displacements, $e(u) = \frac{1}{2}(\nabla u + \nabla u^T)$ denotes the strain tensor and K is the material elasticity tensor (see [162]). Furthermore, under suitable conditions on the data (which are assumed to be satisfied in the rest of the paper), Problem (7.2.1) has a unique weak solution (see [50] for more details).

We consider the following function

$$\begin{aligned} \rho : \Omega &\rightarrow [\rho_{\min}, \rho_{\max}] \\ x &\mapsto \rho(x) \end{aligned}, \quad (7.2.2)$$

which measures the density (i.e., amount) of material at each point of Ω (see [20, 247]), considering a maximum and minimum material density of $\rho_{\max} > 0$ and $\rho_{\min} > 0$, respectively. The total amount of material in Ω , defined by $\int_{\Omega} \rho(x) dx$, must satisfy

$$0 < \int_{\Omega} \rho(x) dx = V_{\text{cons}}. \quad (7.2.3)$$

We assume that K depends on the density ρ in the following way (see [20])

$$K = \rho(x)^p K^0,$$

where $p \geq 1$ is the penalization power which intends to increase the areas of low (close to ρ_{\min}) or high (close to ρ_{\max}) density and reduce the medium (between ρ_{\min} and ρ_{\max}) density zones, as the material is assumed to be isotropic, K^0 is a matrix of the form $K_{i,j,k,l}^0 = 2\lambda\delta_{i,k}\delta_{j,l} + \mu\delta_{i,j}\delta_{k,l}$ where $\delta_{i,j}$ denotes the Kronecker symbol and $\lambda, \mu > 0$ are the Lamé constants of the material.

Following [20, 53], we define the functionals

$$A[\rho](u, v) := \int_{\Omega} K_{i,j,k,l} e_{i,j}(u) e_{i,j}(v) \, dx, \quad (7.2.4)$$

$$l(v) := \int_{\Omega} f \cdot v \, dx. \quad (7.2.5)$$

Then, the weak solution of (7.2.1) is the vector $u \in H = \{u \in [H^1(\Omega)]^d \mid u|_{\Gamma_u} = 0\}$ ($H^1(\Omega)$ is the Sobolev space of all $v \in L^2(\Omega)$ with $\partial_{x_i} v \in L^2(\Omega)$) satisfying $A[\rho](u(\rho), v) = l(v), \forall v \in H$. Thus, the well known minimum compliance design problem can be stated as following

$$\min_{\rho \in E} l(u(\rho)), \quad (7.2.6)$$

$$\text{such that: } A[\rho](u(\rho), v) = l(v), \quad \text{for all } v \in H, \quad (7.2.7)$$

where E is the space of functions satisfying (7.2.2) and (7.2.3) and $u(\rho)$ denotes the (unique) weak solution of (7.2.1).

Here, our purpose is to find the optimal distribution of material ρ when the external load force has a stochastic behavior. In an analogous way to previous stochastic results for truss optimization (see [5, 43]), we assume that the external load force f is randomly perturbed by ξ , with $E(\xi) = 0$. The stochastic topology design problem can be stated as

$$\min_{\rho \in E} \{\alpha E[\Psi(\xi, \rho)] + \beta \text{Var}[\Psi(\xi, \rho)]\} \quad (7.2.8)$$

where the functional Ψ is defined by

$$\Psi(\xi, \rho) = \left\{ \int_{\Omega} (f + \xi) \cdot u \, dx \mid u \in H \text{ satisfies:} \right. \\ \left. A[\rho](u, v) = \int_{\Omega} (f + \xi) \cdot v \, dx \text{ for all } v \text{ in } H \right\} \quad (7.2.9)$$

, $E(\cdot)$ and $\text{Var}(\cdot)$ denote the expected value and the variance of the corresponding random function, and α and $\beta \in \mathbb{R}$.

In this work, we first show that (7.2.8) can be rewritten as a multiload problem and thus solved by using optimization algorithms. Then, we illustrate the interest of this approach by considering a 3D benchmark problem and by comparing the obtained results with those given considering Problem (7.2.6)–(7.2.7).

7.2.2 Variance-expected compliance approach for topology optimization

In this section, we study the stochastic topology design Problem (7.2.9) presented in the introduction. We show that this problem can be transformed into a multiload problem in which the loading scenarios are related to the variance of a random load applied to the body Ω .

In the following, we consider the set $\{P_i\}_{i=1}^{\infty}$ of functions of the Hilbert space $L^2(\Omega)^d$, corresponding to directions of perturbation of the main force f . Also, \mathcal{B} denotes the space of probability and \mathbb{P} its probability measure.

Lemma 29 *Let $\xi: \Omega \times \mathcal{B} \rightarrow \mathbb{R}^d$ be a random load, which in terms of the directions of perturbation $\{P_i\}_{i=1}^\infty$ is written as $\xi = \sum_{i=1}^\infty \varepsilon_i P_i$ where $(\varepsilon_i)_{i=1}^\infty \subset \mathbb{R}$ are random variables with $\mathbb{E}(\varepsilon_i) = 0$ and $\mathbb{E}(\varepsilon_i \varepsilon_j) = \alpha_{i,j}$ for $i, j = 1, \dots, \infty$. Let G be a linear functional. Then*

$$\mathbb{E} \left(\int_{\Omega} \xi \cdot G(\xi) \, dx \right) = \sum_{i=1}^{+\infty} \sum_{j=1}^{+\infty} \alpha_{i,j} \int_{\Omega} P_i \cdot G(P_j) \, dx. \quad (7.2.10)$$

Proof: using $\xi = \sum_{i=1}^{+\infty} \varepsilon_i P_i$ and Fubini's Theorem we obtain

$$\begin{aligned} \int_{\Omega} \int_{\mathbb{R}^d} \xi \cdot G(\xi) \, d\mathbb{P} \, dx &= \int_{\Omega} \int_{\mathbb{R}^d} \sum_{i=1}^{+\infty} \varepsilon_i P_i \cdot G\left(\sum_{j=1}^{+\infty} \varepsilon_j P_j\right) \, d\mathbb{P} \, dx \\ &= \int_{\Omega} \int_{\mathbb{R}^d} \sum_{i=1}^{+\infty} \sum_{j=1}^{+\infty} \varepsilon_i \varepsilon_j P_i \cdot G(P_j) \, d\mathbb{P} \, dx \\ &= \int_{\Omega} \sum_{i=1}^{+\infty} \sum_{j=1}^{+\infty} \left(\int_{\mathbb{R}^d} \varepsilon_i \varepsilon_j \, d\mathbb{P} \right) P_i \cdot G(P_j) \, dx. \end{aligned}$$

□

The following theorem states the relation between the multiload truss model and Problem (7.2.9).

Theorem 30 *Let us consider $\xi: \Omega \times \mathcal{B} \rightarrow \mathbb{R}^d$ be a random load, which in terms of the directions $\{P_i\}_{i=1}^\infty$ is written as $\xi = \sum_{i=1}^\infty \varepsilon_i P_i$ where $(\varepsilon_i)_{i=1}^\infty$ are independent random variables, $\mathbb{E}(\varepsilon_i \varepsilon_j) = 0$ for $i \neq j$, with $\mathbb{E}(\varepsilon_i) = 0$ and $\text{Var}(\varepsilon_i) = \sigma_i^2$. Then the stochastic problem defined in (7.2.8), with $(\alpha, \beta) = (1, 0)$, can be rewritten as the multiload problem:*

$$\min_{\rho \in E} \int_{\Omega} f \cdot u \, dx + \sum_{i=1}^{+\infty} \int_{\Omega} \sigma_i P_i \cdot U_i \, dx \quad (7.2.11)$$

$$A(\rho)[u, v] = \int_{\Omega} f \cdot v \, dx, \quad \forall v \in H \quad (7.2.12)$$

$$A(\rho)[U_i, v] = \int_{\Omega} \sigma_i P_i \cdot v \, dx, \quad \forall v \in H, \forall i \in \mathbb{N} \quad (7.2.13)$$

$$u \in H, U_i \in H \quad \forall i \in \mathbb{N}. \quad (7.2.14)$$

Proof: Let $\rho \in E$ be a feasible material distribution and let us consider $\xi: \mathbb{R}^d \times \mathcal{B} \rightarrow \mathbb{R}$ be a random load with $\mathbb{E}(\xi) = 0$. We define the inverse functional

$$G(f + \xi) = u$$

where u is the unique weak solution of the system (7.2.1) changing f by $f + \xi$. The expected value of the compliance is given by

$$\mathbb{E}(\Psi(\xi, \rho)) = \int_{\mathbb{R}^d} \int_{\Omega} (f + \xi) \cdot u \, dx \, d\mathbb{P}.$$

Using Fubini's theorem and linearity of inverse operator G we have

$$\begin{aligned}
\mathbf{E}(\Psi(\xi, \rho)) &= \int_{\Omega} \int_{\mathbb{R}^d} (f + \xi) \cdot G(f + \xi) \, d\mathbb{P} \, dx \\
&= \int_{\Omega} \int_{\mathbb{R}^d} f \cdot G(f) \, d\mathbb{P} \, dx + \int_{\Omega} \int_{\mathbb{R}^d} \xi \cdot G(f) \, d\mathbb{P} \, dx \\
&\quad + \int_{\Omega} \int_{\mathbb{R}^d} f \cdot G(\xi) \, d\mathbb{P} \, dx + \int_{\Omega} \int_{\mathbb{R}^d} \xi \cdot G(\xi) \, d\mathbb{P} \, dx \\
&= \int_{\Omega} \mathbb{P}(\mathbb{R}^d) f \cdot G(f) \, dx + \int_{\Omega} \mathbf{E}(\xi) G(f) \, dx + \int_{\Omega} G^*(f) \cdot \mathbf{E}(\xi) \, dx \\
&\quad + \int_{\Omega} \int_{\mathbb{R}^d} \xi \cdot G(\xi) \, d\mathbb{P} \, dx \\
&= \int_{\Omega} f \cdot G(f) \, dx + \int_{\Omega} \int_{\mathbb{R}^d} \xi \cdot G(\xi) \, d\mathbb{P} \, dx,
\end{aligned}$$

where G^* denotes the adjoint operator of G .

Using Lemma 29, and the fact that $\mathbf{E}(\varepsilon_i \varepsilon_j) = 0$ for $i \neq j$, we obtain

$$\int_{\Omega} \int_{\mathbb{R}^d} \xi \cdot G(\xi) \, d\mathbb{P} \, dx = \int_{\Omega} \sum_{i=1}^{+\infty} \sigma_i^2 P_i \cdot G(P_i) \, dx.$$

Denoting by U_i the unique weak solution of

$$A(\rho)[U_i, v] = \int_{\Omega} \sigma_i P_i \cdot v \, dx \text{ for all } v \in H,$$

we finally get

$$\mathbf{E}(\Psi(\xi, \rho)) = \int_{\Omega} f \cdot u \, dx + \int_{\Omega} \sum_{i=1}^{+\infty} \sigma_i P_i \cdot U_i \, dx.$$

□

Theorem 30 is useful because, when its assumptions are satisfied, it gives an explicit expression of Problem (7.2.8) easy to evaluate. In general cases, it could be difficult to evaluate directly $\mathbf{E}[\Psi(\xi, \rho)]$. We can consider, for instance, a *Monte-Carlo* algorithm (see [136]) to approximate those values. However, this method, and thus the resolution of (7.2.8), is numerically expensive.

We can now introduce the following result:

Lemma 31 *Using the notation of the previous theorem, let us consider $\xi = \varepsilon P$ be a random perturbation of f . The vector $P \in L^2(\Omega)^d$ and ε is a random variable with moments $\mathbf{E}(\varepsilon^i) = \mu_i$ (of course we assume $\mu_1 = 0$). Then*

$$\begin{aligned}
\text{Var} \left(\int_{\Omega} (f + \xi) \cdot G(f + \xi) \, dx \right) &= \mu_2 \left[\int_{\Omega} P \cdot G(f) \, dx + \int_{\Omega} f \cdot G(P) \, dx \right]^2 \\
&\quad + 2\mu_3 \int_{\Omega} P \cdot G(P) \, dx \left[\int_{\Omega} P \cdot G(f) \, dx + \int_{\Omega} f \cdot G(P) \, dx \right] \\
&\quad + (\mu_4 - \mu_2) \left(\int_{\Omega} P \cdot G(P) \, dx \right)^2 \quad (7.2.15)
\end{aligned}$$

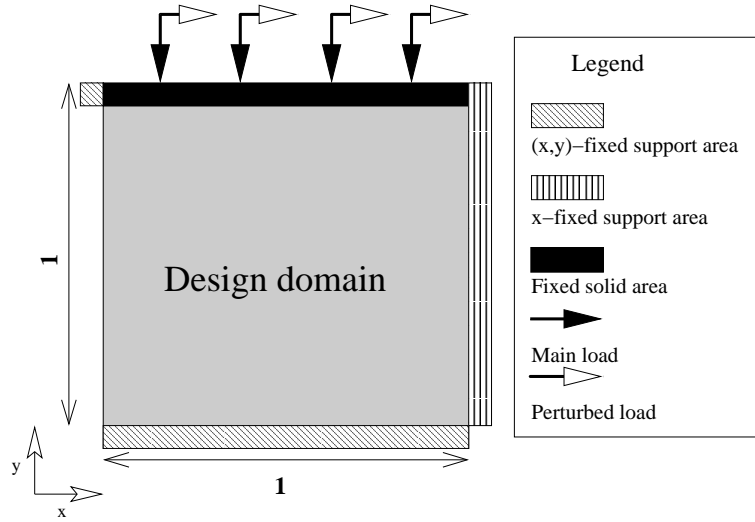


Figure 7.9: Geometrical representation of the design problem described in Section 7.2.3: design domain (gray), support areas (dashed), fixed solid area (black) and loads (arrow) of the considered numerical example.

The proof is left to the reader.

Thus, assuming that $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ (in this case $\mu_1 = 0, \mu_2 = \sigma^2, \mu_3 = 0$ and $\mu_4 = 3\sigma^4$) and considering the Equations (7.2.10) and (7.2.15), we can consider for the topology optimization case the problem (7.2.8) with $\beta \neq 0$.

In Sections 7.2.3-7.2.4, we present 2D and 3D numerical experiments used to illustrate the interest of formulation (7.2.8). We note that the 3D case is currently a work in progress and only results corresponding to $(\alpha, \beta) = (1, 0)$ are presented.

7.2.3 2D Numerical example

Problem description

In order to perform a first numerical study to see the interest of (7.2.8) and the choice of the parameters (α, β) , we first consider a 2-D benchmark design problem. A geometrical representation of this benchmark problem is given by Figure 7.9. We are interested in designing a 'bridge' considering various fixed support areas. The upper part of the bridge (black part in Figure 7.9) is also fixed with a density equal to 1 and a main load $f = (0, -1)$ is applied homogeneously all over this part. Since we are interested in generating structures that are stable to perturbations of the design force f , we consider a random load $\xi = \xi_1 V_1$, with ξ_1 of law $\mathcal{N}(0, 1)$ and $V_1 = (1, 0)$, applied at the same region than f .

We solve the corresponding problem (7.2.8), considering a finite element approach similar to the one proposed in [247] with a grid of 30×30 , for various set of values of (α, β) . More precisely, (α, β) is chosen in the set $\Sigma = \{(1, 0), (1, 0.5), (1, 1), (0.5, 1), (0, 1)\}$. In addition to those problems, we also solve the same design problem considering the perturbation load as the main load and omitting f , with $(\alpha, \beta) = (1, 0)$. In order to solve (7.1.13) for the considered design problems, we consider the SDMSA-L2, presented

Table 7.2: Results obtained considering $\{\rho_{(\alpha,\beta)}\}_{(\alpha,\beta)\in\Sigma}$ and $\rho_{\text{Pert.}}$: Expected compliance (**EC**), Variance (**Vari**) and Coherent Value at Risk (**C-VaR₁**) of the solutions (**Sol.**). The value in parenthesis represents the difference in percentage between the values of the considered solution and $\rho_{(1,0)}$.

Sol.	$\rho_{(1,0)}$	$\rho_{(1,0.5)}$	$\rho_{(1,1)}$	$\rho_{(0.5,1)}$	$\rho_{(0,1)}$	$\rho_{\text{Pert.}}$
EC	2.6	2.7 (+2%)	2.7 (+4%)	2.8 (+6%)	4.6 (+73 %)	5.2 (+97%)
Vari	4.4	4.1 (-6%)	3.0 (-30%)	2.9 (-35%)	2.5 (-42%)	2.7 (-38%)
C-VaR₁	13.2	12.0 (-9%)	11.9 (-10%)	11.6 (-12%)	13.5 (+1%)	13.8 (+2%)

in Section 2.3.1 (with $A_2, t_0 = 50, t_{l_1} = 5, t_{l_2} = 5$ and $\epsilon = 0$). The obtained solutions are denoted by $\{\rho_{(\alpha,\beta)}\}_{(\alpha,\beta)\in\Sigma}$ for the full design problem and $\rho_{\text{Pert.}}$ for the only perturbation load case.

In order to have a qualitative comparison of those structures, we analyze their robustness when they are submitted to random loads and their shape evolution. More precisely, for each solution ρ , we consider the random variable $\Phi_\rho = \Psi(\xi, \rho)$, where Ψ is defined by (7.1.12), and we approximate its density function ρ_{Φ_ρ} using a Monte-Carlo approach [136] that generates $M \in \mathbb{N}$ possible *scenarios* (i.e. values of ξ). Then, we compute the *Coherent-Value at Risk* (C-VaR) of the structure as defined in Section 7.1.4.

Results

All results are reported in Table 7.2 and the different shape configurations are presented in Figure 7.4.

As we can observe in Table 7.2, the solution $\rho_{(1,0)}$ is less stable to perturbations of the main load, considering the 1 % worst case scenarios (C-VaR₁), than the solutions $\rho_{(1,0.5)}, \rho_{(1,1)}, \rho_{(0.5,1)}$. Although the value of the expected compliance increases as the value of β increases, those three structures represent a good alternative to $\rho_{(1,0)}$. In fact the increase of the expected compliance is reasonable (between 2 and 6%) in comparison to the C-VaR₁ diminution (between 9 and 12%). However, considering only a variance minimization problem generates the structure with worst characteristics.

Figures 7.10 shows that the shape of the structure changes with the evolution of the coefficients α and β . In fact, it seems that there is a mass transfer phenomenon when β goes from 0 to 1: From the right initial pylon to the creation of the left inclined pylon. Furthermore, the shape $\rho_{(0,1)}$ is close to the $\rho_{\text{Pert.}}$ ones. This shape evolution is intuitive as the right pylon provides a good resistance to the vertical load f while the inclined pylons support inclined loads resulting of the perturbation of f .

As in the truss optimization case, we deduce that considering formulation (7.1.13) for topology optimization with an adequate balance between α and β helps to generate structures more robust to perturbations of the main load.

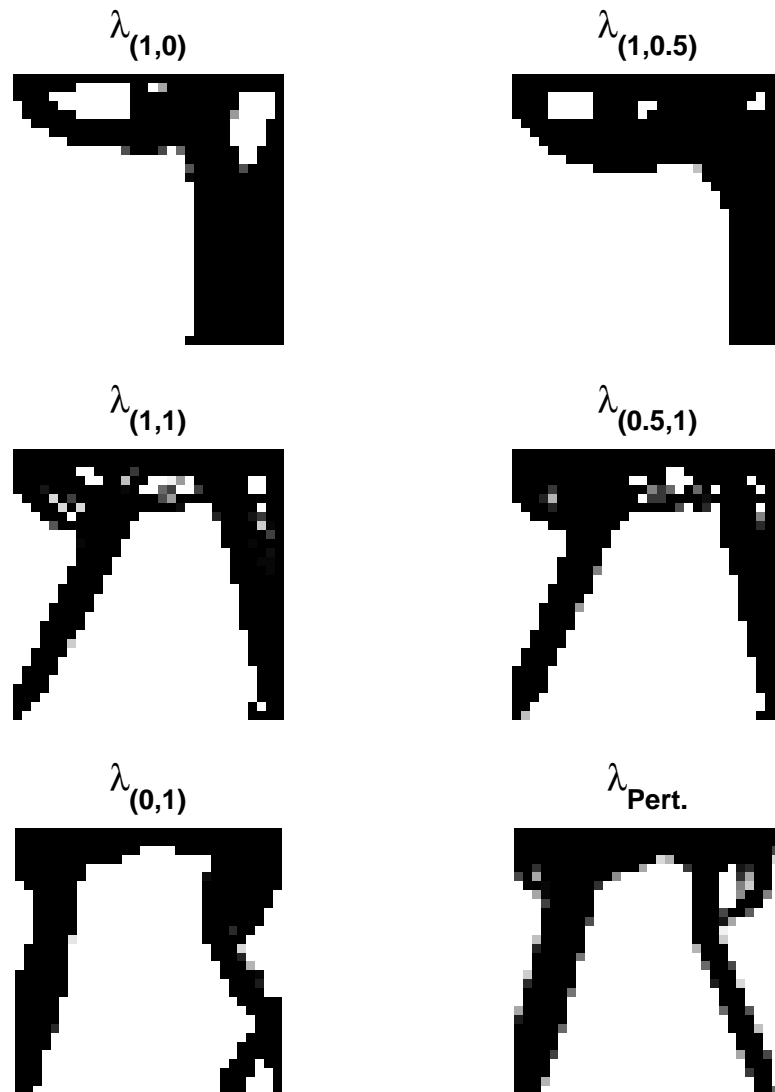


Figure 7.10: Shape of the solutions $\{\rho_{(\alpha,\beta)}\}_{(\alpha,\beta)\in\Sigma}$ and $\rho_{\text{Pert.}}$.

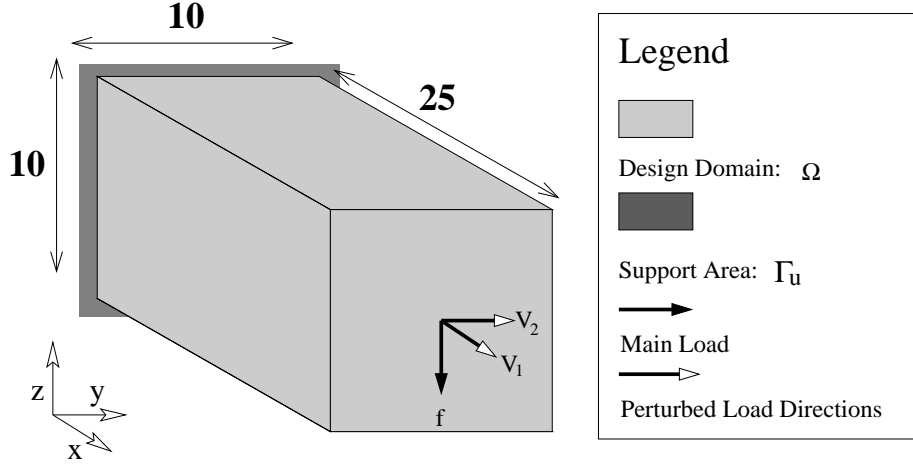


Figure 7.11: Geometrical representation of the 3D cantilever design problem described in Section 7.2.4: design domain Ω (gray), support area Γ_u (dark gray), main load f and perturbed load directions V_1 and V_2 (arrows).

7.2.4 3D Numerical example

Problem description

We consider a 3D benchmark design problem that consists in designing a cantilever submitted to a vertical load [20]. More precisely, we consider a rectangular domain $\Omega = [0, 25] \times [0, 10] \times [0, 10]$, compound by a material with a minimum and maximum density of $\rho_{\min} = 10^{-3}$ and $\rho_{\max} = 10$, respectively. The face $\{0\} \times [0, 10] \times [0, 10]$ of Ω is fixed to a support area. The total amount of material is $V_{\text{cons}} = 1250$. A main point load $f = (0, 0, -1)$ is applied at the node $(25, 5, 5)$. We consider a random point load $\xi = \xi_1 V_1 + \xi_2 V_2$, with ξ_1 and ξ_2 random variables of law $\mathcal{N}(0, 25)$, $V_1 = (1, 0, 0)$ and $V_2 = (0, 1, 0)$, applied at the same node than f . We consider the typical value $p = 3$ [247]. A geometrical representation of this benchmark problem is given in Figure 7.11.

We want to solve Problems (7.2.6)–(7.2.7) and (7.2.11)–(7.2.14) associated to the cantilever. To do so, we consider a finite element method, similar to the one proposed in [247], with a discretization given by $N_x = 25$, $N_y = 10$ and $N_z = 10$, where N_x , N_y and N_z are the number of equispaced elements in the X, Y and Z directions, respectively. Thus, $N_{\text{el}} = N_x \times N_y \times N_z = 2500$ is the total number of elements used. Then, both optimization problems are solved by using the SDMSA-L2, presented in Section 2.3.1 (with $A_2, t_0 = 50$, $t_{l_1} = 5$, $t_{l_2} = 5$ and $\epsilon = 0$). The obtained solutions are denoted by ρ_{sp} for Problem (7.2.6)–(7.2.7) and ρ_{ec} for Problem (7.2.11)–(7.2.14).

As done in Section 7.2.3, in order to have a qualitative comparison of ρ_{sp} and ρ_{ec} , we analyze their robustness when they are submitted to random loads and their density distribution. For this purpose, we first compute the compliance value without considering any perturbed load. Then, for each solution $\rho \in \{\rho_{\text{sp}}, \rho_{\text{ec}}\}$, we consider the random variable $\Phi_\rho = \Psi(\xi, \rho)$. We approximate the density function of Φ_ρ , denoted by γ_{Φ_ρ} , by using a Monte-Carlo approach [136] that generates $M \in \mathbb{N}$ possible *scenarios* (i.e., values of ξ). Then, we calculate some statistical values of γ_{Φ_ρ} : its mean, maximum and

Solution	Compl	EC	C-VaR₁	Max
ρ_{sp}	1683	44798	321612	415021
ρ_{ec}	3906	11129	61487	103936
Percent variation	+ 132	-75	-81	-75

Table 7.3: Results obtained when considering ρ_{sp} and ρ_{ec} : Compliance value (**Compl**), expected compliance (**EC**), coherent value at risk (**C-VaR₁**) of Φ_ρ and maximum value (**Max**) of Φ_ρ . The last line corresponds to the percent variation (%) between the results given by ρ_{sp} and ρ_{ec} .

1%-Coherent-Value at Risk (C-VaR₁) values.

Results

Some results found with the numerical experiments presented in Section 7.2.4 are reported in Table 7.3. The density distribution of ρ^{sp} and ρ^{ec} are presented in Figures 7.12 and 7.13, respectively.

As we can see on Table 7.3, the solution ρ_{sp} has a non perturbed compliance value of 1683. The compliance value of ρ_{ec} is 3906 (more than twice higher). However, when considering perturbed loads, the expected compliance, the C-VaR₁ and the maximum values of $\Phi_{\rho_{ec}}$ are more than four times lower than those of $\Phi_{\rho_{sp}}$. This result is expected as the distribution ρ_{sp} is adapted to resist to the main load but is less stable to perturbations of f . In counterpart, ρ_{ec} allows to obtain reduced perturbed compliance values. This can also be deduced by observing the shape of both solutions (see Figures 7.12–7.13): the density distribution ρ_{sp} is concentrated in the X-Z plane including f , producing a good resistance to the main load, whereas ρ_{ec} exhibits four pillars resilient to the perturbed loads included in the X-Y plane orthogonal to f .

From this preliminary result, as in the truss optimization case studied in [43], we can deduce that considering formulation (7.2.11) for topology optimization can help to generate structures robust to perturbations of the main load.

7.2.5 Conclusions and Perspectives

In this Section, We have adapted an expected compliance formulation, coming from truss design (see Section 7.1 and [5, 43]), to topology optimization. We have exhibited, in a particular case, an explicit expression easy to evaluate. This method has been tested numerically on a 2D benchmark experiment for the Variance-Expected Compliance case and a 3D benchmark experiment for the Expected Compliance case. This approach allows to generate structures that are more stable to perturbations of the main load than considering a classical non perturbed load approach.

As a work in progress, the results given in this Section will be extended by performing 3D numerical experiments for the Variance-Expected Compliance model.

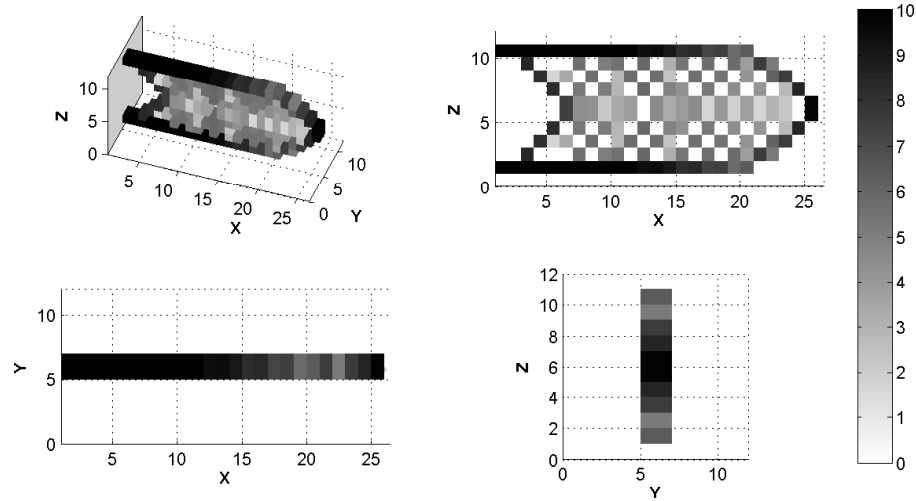


Figure 7.12: Shape and density distribution of ρ_{sp} : **(Top-Left)** Perspective view with representation of the support area (gray plane), **(Top-Right)** Lateral view (i.e., X - Z plane), **(Bottom-Left)** Top view (i.e., X - Y plane) and **(Bottom-Right)** Face view (i.e., Y - Z plane). The gray scale color map representing the density is also presented.

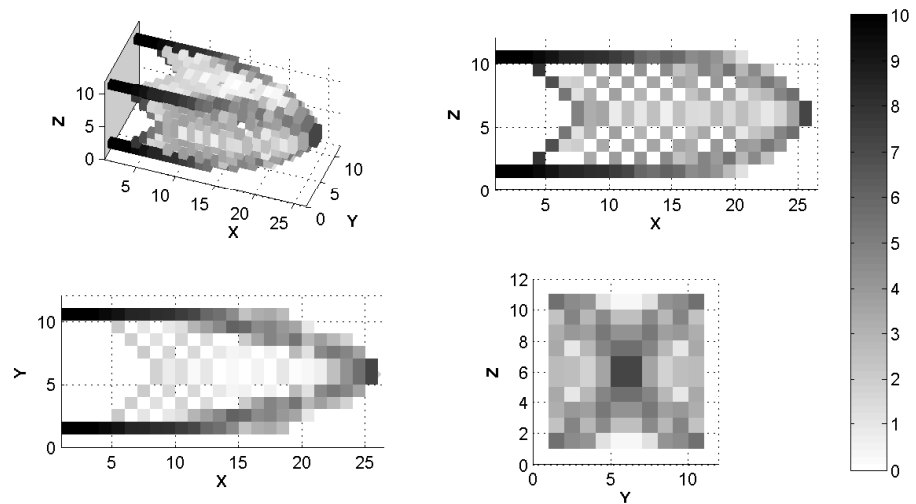


Figure 7.13: Shape and density distribution of ρ_{ec} : **(Top-Left)** Perspective view with representation of the support area (gray plane), **(Top-Right)** Lateral view (i.e., X - Z plane), **(Bottom-Left)** Top view (i.e., X - Y plane) and **(Bottom-Right)** Face view (i.e., Y - Z plane). The gray scale color map representing the density is also presented.

Chapter 8

Conclusions Générales

Nous avons présenté tout au long de ce mémoire mes principaux axes de recherche étudiés après ma thèse. Nous avons pu apprécier, du Chapitre 3 au 7, la diversité des problèmes traités (épidémiologie, traitement alimentaire, dépollution, télécommunication, etc.) et leur utilité dans la société actuelle (économique, écologique, sanitaire, ingénierie, etc.). Afin d'analyser et de tenter de résoudre ces problèmes nous avons eu recours à diverses méthodes et techniques provenant de la modélisation mathématique (EDPs, EDOs, stochastique, etc.), de la simulation numérique (schémas en volumes finis, en éléments finis, de Monte-Carlo, etc.) et de l'optimisation (en particulier les algorithmes introduits dans le Chapitre 2).

Les résultats, conclusions et possibles perspectives de recherche spécifiques à chaque travail ont été détaillés dans leur Chapitre respectif. De manière générale, la plupart des thèmes abordés ont permis d'obtenir des résultats satisfaisants (publiés dans des journaux internationaux) et ont ouvert de nouvelles possibilités de recherche. Certaines étant actuellement en cours d'études (tels que l'ajout d'un modèle économique à Be-FAST, l'analyse de nouveaux schémas de discrétisation du modèle du bateau 'Skimmer', etc.) et d'autres en projet pour le futur (telles que l'optimisation des mesures pour contrôler une épidémie de PPC, l'étude d'un traitement alimentaire à base de pression température afin de désactiver plusieurs types de protéines, etc.).

Comme remarque finale, il est à noter que seuls les travaux dans un état de développement avancé ont été présentés dans ce document. Certaines problématiques annexes n'ont pas été abordées:

- L'étude de sensibilité du modèle à temps de défaut présenté en thèse pour optimiser un portefeuille de crédit. Prépublication disponible:
 - *R. Abella Muñoz, I. Armero Huertas, B. Ivorra and A. M. Ramos. Sensitivity analysis of a default time model for credit risk portfolio management. Prépublications du Departamento de Matemática Aplicada - UCM - N° MA-UCM-2010-16.*
- Méthodes numériques pour résoudre un jeu différentiel duopolistique avec un contrôle en boucle fermée. *En cours de rédaction.*
- Adaptation de Be-FAST au cas d'une épidémie de PPC en Bulgarie. *En cours de rédaction.*

Bibliography

- [1] W. Aichtziger (1997) Topology optimization of discrete structures: an introduction in view of computational and nonsmooth aspects. In *Topology optimization in structural mechanics*, volume 374 of *CISM Courses and Lectures*, pages 57–100. Springer, Vienna.
- [2] W. Aichtziger (1998) Multiple-load truss topology and sizing optimization: some properties of minimax compliance. *J. Optim. Theory Appl.* 98(2), 255–280
- [3] W. Aichtziger, M. Bendsøe, A. Ben-Tal, and J. Zowe (1992) Equivalent displacement based formulations for maximum strength truss topology design. *Impact Comput. Sci. Engrg.*, 4(4):315–345.
- [4] C. Alavani, R. Glowinski, S. Gomez, B. Ivorra, P. Joshi, A.M. Ramos, Modelling and simulation of a polluted water pumping process, *Mathematical and Computer Modelling*, 51 (2010), 461-472.
- [5] F. Alvarez and M. Carrasco (2005) Minimization of the expected compliance as an alternative approach to multiload truss optimization. *Struct. Multidiscip. Optim.*, 29(6):470–476.
- [6] G. Allaire, A. Craig. *Numerical Analysis and Optimization: An Introduction to Mathematical Modelling and Numerical Simulation*. Oxford University Press, USA. (2007).
- [7] G. Allaire. *Shape Optimization by the Homogenization Method*. Springer. (2002).
- [8] R. Anderson, May, R. (1979). Population biology of infectious diseases: Part 1, *Nature* 280: 361–367.
- [9] A.P.G. Depollution, Website: <http://apgdepollution.free.fr>
- [10] R. Aris, *Vectors, Tensors, and the Basic Equations of Fluid Mechanics*. Dover Publications, Inc. New York, (1989).
- [11] P. Artzner, Delbaen, D., Eber, J.M., Heath, D.: Coherent measures of risk. *Mathematical Finance*. 9: 203–228 (1999).
- [12] P. Artzner, Delbaen, D., Eber, J.M., Heath, D.: Thinking coherently. *Risk*. 10: 68–71 (1997)

-
- [13] H. Attouch and R. Cominetti. A dynamical approach to convex minimization coupling approximation with the steepest descent method. *Journal of Differential Equations*, 128(2):519–540, (1996).
- [14] H. Attouch, X. Goudou, and P. Redont. The heavy ball with friction method. I: The continuous dynamical system: Global exploration of the local minima of a real-valued function by asymptotic analysis of a dissipative dynamical system. *Commun. Contemp. Math.*, 2(1):1–34, (2000).
- [15] C. Audet and J.E. Dennis Jr. "Analysis of Generalized Pattern Searches." *SIAM Journal on Optimization*. 13(3): 889–903. (2003).
- [16] A. Baeza, C. Castro, F. Palacios, E. Zuazua. 2-D Euler Shape Design on Nonregular Flows Using Adjoint RankineHugoniot Relations. *AIAA Journal*. 47(3):552–562. (2009).
- [17] Y. Bai, de Klerk, E., Pasechnik, D., Sotirov, R.: Exploiting group symmetry in truss topology optimization. *Optim. and Eng.* 10(3): 331–349 (2009)
- [18] R. Baxter (1982) *Exactly Solved Models in Statistical Mechanics*. Academic Press
- [19] A. Ben-Tal and A. Nemirovski (1997) Robust truss topology design via semidefinite programming. *SIAM J. Optim.*, 7(4):991–1016.
- [20] M. P. Bendsøe and O. Sigmund (2003) *Topology optimization. Theory, methods and applications*. Springer-Verlag, Berlin.
- [21] A. Ben-Tal and M. Zibulevsky (1997) Penalty/barrier multiplier methods for convex programming problems. *SIAM J. Optim.*, 7(2):347–366.
- [22] A. Ben-Tal, Bendsøe, M.P.: A new method for optimal truss topology design. *SIAM J. Optim.* 3(2): 322–358 (1993)
- [23] J.M. Berg, J.L. Tymoczko, and L. Stryer. *Biochemistry (5th edition)*. W.H. Freeman, New York, 2002.
- [24] T. Back, D.B. Fogel, Z. Michalewicz, (eds.) *Evolutionary computation 1: basic algorithms and operators*, IOP Publishing, Bristol, (2000).
- [25] S.A. Berger, L. Talbot, and L.S. Yao. Flow in curved pipes. *Annual Review of Fluid Mechanics*, 15(1):461–512, (1983).
- [26] A. Bermúdez de Castro, Gómez, D. Quintela, P. Salgado, P. (Eds.). *Numerical Mathematics and Advanced Applications*. Springer. (2006).
- [27] R. Best and G. Hummer. Coordinate-dependent diffusion in protein folding. *Proc Natl Acad Sci U S A*, 107:1088–93, Jan (2010).
- [28] J. C. W. Berkhoff. Computation of combined refraction-diffraction. In ASCE., editor, *Proc. 13th Coastal Eng. Conf., Vancouver*, pages 471–490, (1972).

-
- [29] C. Bock, Prat J (2005) Scalable wdma?tdma protocol for passive optical networks that avoids upstream synchronization and features dynamic bandwidth allocation. *J Opt Netw* 4(4):226–236
- [30] H.G. Bock; Kostina, E.; Hoang, X.P.; Rannacher, R. (Eds.). *Modeling, Simulation and Optimization of Complex Processes*. Springer. (2005).
- [31] N. Booij. Gravity waves on water with non-uniform depth and current. (*PhD thesis*) *Technical University of Delft, The Netherlands*, (1981).
- [32] F. Brauer & Castillo-Chavez, C. (2001). *Mathematical Models in Population Biology and Epidemiology*, Springer.
- [33] Y. Braverman, Rechtman, S., Frish, A., Braverman, R. Dynamics of biting activity of *C. Imicola kieffer* (diptera:Ceratopogonidae) during the year. *Veterinary Medicine*, 58(2-3), (2003).
- [34] H. Brezis. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Springer. (2010).
- [35] R. Bruyere. *Credit Derivatives and Structured Credit*. Wiley Finance. (2005).
- [36] J.P. Brody P., Yager, R.E. Goldstein, and R.H. Austin. Biotechnology at low reynolds numbers. *Biophysical journal*, 71(6):3430–3441, (1996).
- [37] A.A. Brown, M.C. Bartholomew-Biggs. Some effective methods for unconstrained optimization based on the solution of systems of ordinary differential equations. *Journal of Optimization Theory and Applications*, 62(2):211–224, (1989).
- [38] R.A.R. Butler, E.E. Slaminka. An evaluation of the sniffer global optimization algorithm using standard test functions. *J. Comput. Phys.* 99(1): 28–32, (1992).
- [39] A. Caflisch and E. Paci. *Molecular Dynamics Simulations to Study Protein Folding and Unfolding - Protein Folding Handbook*, pages 1143–1169. Wiley-VCH Verlag GmbH, (2008).
- [40] M.P. Calvo, J. de Frutos, J. Novo, Linearly implicit Runge-Kutta methods for advection-reaction-diffusion equations, *Applied Numerical Mathematics* 37(4) (2001) 535–549.
- [41] M. Carrasco. Diseño óptimo de estructuras reticulares en elasticidad lineal vía teoría de la dualidad. Estudio teórico y numérico. Engineering Degree Thesis, Universidad de Chile (2003)
- [42] M. Carrasco, Ivorra, B., Lecaros, R., Ramos, A.M. A variance-expected compliance approach for topology optimization. In: Hélder, R., Herskovits, J. (eds.). *CD-ROM Proceedings of the ENGOPT* (2010). Instituto Superior Técnico, Lisboa (2010)

-
- [43] M. Carrasco, B. Ivorra and A.M. Ramos (2011) A variance-expected compliance model for structural optimization. *Journal of Optimization Theory and Applications*, Accepted.
- [44] C.K. Chan, Tong F, Chen LK, Song J, Lam D (1997) A practical passive surveillance scheme for optically amplified passive branched optical networks. *Photonics Technology Letters, IEEE* 9(4):526–52
- [45] A. Chawla, Ozkan-Haller HT, Kirby JT. Spectral model for wave transformation over irregular bathymetry. *Journal of Waterway, Port, Coastal and Ocean Engineering*. (1998). 124:189–198.
- [46] H.C. Cheng, Huang JF, Chen YH (2008) Reconstruction of phase-shifted fiber bragg grating parameters using genetic algorithm over thermally-modulated reflection intensity spectra. *Optical Fiber Technology* 14(1):27–35
- [47] J. D. Chodera, W. C. Swope, J. W. Pitera, C. Seok, and K. A. Dill. Use of the weighted histogram analysis method for the analysis of simulated and parallel tempering simulations. *Journal of Chemical Theory and Computation*, 3(1):26–41, (2007).
- [48] J. Chow, Town G, Eggleton B, Ibsen M, Sugden K, Bennion I (1996) Multiwavelength generation in an erbium-doped fiber laser using in-fiber comb filters. *Photonics Technology Letters, IEEE* 8(1):60–62
- [49] K.P. Chuang, Lai Y, Sheu LG (2004) Pure apodized phase-shifted fiber bragg gratings fabricated by a two-beam interferometer with polarization control. *Photonics Technology Letters, IEEE* 16(3):834–836
- [50] P. Ciarlet (1988) *Mathematical Elasticity, Vol. I, Three Dimensional Elasticity*. North-Holland, Amsterdam.
- [51] D. Cioranescu, J.L. Lions *Nonlinear Partial Differential Equations and Their Applications: Collège de France Seminar, Volumen 14*. Elsevier. (2002).
- [52] R. Cominetti and Alemany O. Steepest descent evolution equations: asymptotic behavior of solutions and rate of convergence, *Transactions of the AMS* 351(12): 4847–4860. (1999).
- [53] S. Conti, H. Held, M. Pach, M. Rumpf and R. Schultz (2008) Shape optimization under uncertainty, a stochastic programming perspective. *SIAM Journal on Optimization*. 19(4):1610–1632
- [54] N. Darnton, O. Bakajin, R. Huang, B. North, J.O. Tegenfeldt, E.C. Cox, J. Sturm, and R.H. Austin. Hydrodynamics in 2-1/2 dimensions: making jets in a plane. *Journal of Physics: Condensed Matter*, 13:4891–4902, (2001).
- [55] W. Dorn, Gomory, R., Greenberg, M.: Automatic design of optimal structures. *J. mécanique*. 3:25–52. (1964).

-
- [56] R. Dautray, J.L. Lions, J.C. Amson, M. Artola, M. Cessenat. *Mathematical Analysis and Numerical Methods for Science and Technology: Volume 3: Spectral Theory and Applications*. Springer. (1990).
- [57] K.A. De Jong, *Evolutionary Computation: a unified approach*. MIT press, (2006).
- [58] D. De Sancho, U. Doshi, and V. Muñoz. Protein folding rates and stability: how much is there beyond size? *J. Am. Chem. Soc.*, 131:2074–5, Feb (2009).
- [59] W.R. Dean. The stream-line motion of fluid in a curved pipe (second paper). *Philosophical Magazine*, 5(30):673–695, (1928).
- [60] R.G. Dean RG and Dalrymple RA. *Water Wave Mechanics for Engineers and Scientists*, vol. 2. World Scientific: Singapore, (1991).
- [61] D. DeAngelis, & Gross, L. (1991). *Individual-based Models and Approaches in Ecology*, Chapman and Hall, NY.
- [62] L. Debiane, Ivorra, B., Mohammadi, B., Nicoud, F., Ern, A., Poinso, T., Pitsch, H.: A low-complexity global optimization algorithm for temperature and pollution control in flames with complex chemistry. *Int. J. of Comput. Fluid Dynamics*, 20(2):93–98 (2006)
- [63] A. Delgado, C. Rauh, W. Kowalczyk and A. Baars, Review of modelling and simulation of high pressure treatment of materials of biological origin, *Trends in Food Science & Technology* 19(6):329–336 (2008).
- [64] S. Denys, A. van Loey and M.E. Hendrickx, A modelling approach for evaluating process uniformity during batch high hydrostatic pressure processing: combination of a numerical heat transfer model and enzyme inactivation kinetics, *Innovative Food Science and Emerging Technologies* 1:5–19 (2000).
- [65] L. DeRose & Padua, D. (1999). Techniques for the translation of Matlab programs into Fortran 90, *ACM Trans. on Progr. Lang. and Syst.* 21(2): 285–322.
- [66] D. Di Serafino, S. Gomez, L. Milano, F. Riccio and G. Toraldo., A genetic algorithm for a global optimization problem arising in the detection of gravitational waves, *Journal of Global Optimization*, 48(1) (2010), 41–55.
- [67] J. I. Díaz, *Nonlinear Partial Differential Equations and Free Boundaries. Vol.I. Elliptic equations*. Research Notes in Mathematics n° 106, Pitman, Londres, (1985).
- [68] J.I. Díaz, S. Shmarev. Lagrangian approach to level sets: application to a free boundary problem arising in climatology. *Archive for Rational Mechanics and Analysis*. 194(1): 75–103. (2009).
- [69] K.D. Dolan, L. Yang and C.P. Trampel, Nonlinear regression technique to estimate kinetic parameters and confidence intervals in unsteady-state conduction-heated foods, *J. Food Eng.* 80:581–593. (2007).

- [70] L. Dumas, V. Herbert, and F. Muyl. Hybrid method for aerodynamic shape optimization in automotive industry. *Computers and Fluids*, 33(5):849–858, 2004.
- [71] J. Dunbar, H.P. Yennawar, S. Banerjee, J. Luo, and G.K. Farber. The effect of denaturants on protein structure. *Protein Science*, 6(8):1727–1733, 1997.
- [72] H. Eckhardt: Kinematic Design of Machines and Mechanisms. McGraw-Hill Prof. (1998).
- [73] R. Eymard, T. Gallouet, R. Herbin and A. Michel, Convergence of a finite volume scheme for nonlinear degenerate parabolic equations, *Numer. Math.*, 92(1) (2002), 41–82.
- [74] S. Edwards, Fukusho, A., Lefevre, P., Lipowski, A., Pejsak, Z., Roehe, P. & Westergaard, J. (2000). CSF: the global situation, *Vet. Microbiol.* 27(3): 103–109.
- [75] S.E. El-khamy, Balamesh AS (1987) Selection of gold and kasami code sets for spread spectrum cdma systems of limited number of users. *International Journal of Satellite Communications* 5(1):23–32
- [76] A. Elbers, Stegeman, A., Moser, H., Ekker, H., Smak, J. & Pluimers, H. (1999). The CSF epidemic 1997-1998. in The Netherlands: descriptive epidemiology, *Prev. Vet. Med.* 4: 157–184.
- [77] T. Erdogan (1997) Fiber grating spectra. *Journal of Lightwave Technology*. 15(8): 1277–1294
- [78] M. Ericsson, , M.G.C. Resende, P.M. Pardalos. A genetic algorithm for the weight setting problem in OSPF routing. *J. Comb. Optim.*, 6(3):299–333, 2002.
- [79] E.S.R.I. (2012). ArcGIS. Website: <http://www.esri.com/software/arcgis/>
- [80] M. Etezad, Kahrizi M, Khorasani K (2011) A novel multiplexed fiber bragg grating sensor with application in structural health monitoring. *physica status solidi (c)* 8(9):2953–2956
- [81] X. Fang, Wang DN, Li S (2003) Fiber bragg grating for spectral phase optical code-division multiple-access encoding and decoding. *J Opt Soc Am B* 20(8):1603–1610
- [82] V. Faber, and Manteuffel T., Necessary and Sufficient Conditions for the Existence of a Conjugate Gradient Method, *SIAM J. Numer. Anal.* 21 (1984) 315–339.
- [83] E. Fernández, Ivorra, B., Ramos, A., Martínez-Lopez, B. & Sánchez-Vizcaíno, J. (2011). Diseño de un modelo económico y de planes de control para una epidemia de Peste Porcina Clásica, *Preprint de la Universidad Complutense de Madrid*. Available at: <http://www.ucm.es/centros/cont/descargas/documento28250.pdf>
- [84] T.A. Feo, M.G.C. Resende, Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6:109–133, (1995).

-
- [85] T.A. Feo, K. Sarathy, J. McGahan A grasp for single machine scheduling with sequence dependent setup costs and linear delay penalties . *Computers & Operations Research* 23(9):881–895, (1996).
- [86] A.J.M. Ferreira. *MATLAB Codes for Finite Element Analysis*. Solid Mechanics and its Applications. Springer, (2009).
- [87] D.B. Fogel, An Introduction to Simulated Evolutionary Optimization, IEEE Trans. on Neural Networks: Special Issue on Evolutionary Computation, 5, (1994).
- [88] C. M. Fonseca and J. Fleming. An overview of evolutionary algorithms in multi-objective optimization. *Evolutionary Computation*, 3(1):1–16, (1995).
- [89] R. Fletcher, Conjugate Gradient Methods for Indefinite Systems, in Proc. Dundee Conference on Numerical Analysis, Lecture Notes in Mathematics 506, G. A. Watson, ed., Springer-Verlag, Berlin, pp. 73–89. (1976).
- [90] C.A. Floudas, P.M. Pardalos *A collection of test problems for constrained global optimization algorithms*. In *Lecture Notes in Computer Science*, 455, Springer Berlin Heidelberg New York, (1990).
- [91] C.A. Floudas, P.M. Pardalos, C.S. Adjiman, W.R. Esposito, Z. Gumus, S.T. Harding, J.L. Klepeis, C.A. Meyer, and C.A. Schweiger. *Handbook of test problems in local and global optimization*, Kluwer Academic Publishers, (2000).
- [92] S. Forrest (1993) Genetic algorithms: principles of natural selection applied to computation. *Science* 261(5123):872–878
- [93] A. Fraguera, J.A. Infante, A.M. Ramos and J.M. Rey, Identification of a heat transfer coefficient when it is a function depending on temperature. *WSEAS Trans. Math.* 7(4):160–172. (2008).
- [94] N. Fraysse. Aménagement du lido de Sète à Marseillan. Protection de la partie nord du lido: étude comparative de solution. Technical Report, Bas-Rhône Languedoc Ingénierie (2006).
- [95] R. Freund and N. Nachtigal, QMR: A Quasi-Minimal Residual Method for Non-Hermitian Linear Systems, *Numer. Math.* 60 (1991) 315–339.
- [96] E. Gemzický, Müllerová J (2008) Apodized and chirped fiber bragg gratings for optical communication systems: influence of grating profile on spectral reflectance. *SPIE*, vol 7138, p 71380X
- [97] J. Gloster, Mellor, P.S., Burgin, L., Sanders, C. and Carpenter S., Will bluetongue come on the wind to the United Kingdom in 2007? *Vet.Rec.* 160:422–6. (2007).
- [98] F. Glover. Tabu Search - Part I, *ORSA Journal on Computing*, 1(3): 190–206, (1989).
- [99] F. Glover. Tabu Search - Part II, *ORSA Journal on Computing*, 2(1): 4–32, (1990).

-
- [100] R. Glowinski, P. Neittaanmaki, Partial Differential Equations. Modelling and Numerical Simulation, Series: Computational Methods in Applied Sciences, Springer, 16, (2008).
- [101] N. Gö. Protein folding as a stochastic process. *Journal of Statistical Physics*, 30(2):413–423, (1983).
- [102] D. Goldberg, Genetic algorithms in search, optimization and machine learning, Addison Wesley, (1989).
- [103] G. Golub, J.M. Ortega, Scientific Computing: An Introduction with Parallel Computing, CA: Academic Press, San Diego, (1993).
- [104] S. Gomez, G. Severino, L. Randazzo, G. Toraldo, J.M. Otero., Identification of the hydraulic conductivity using a global optimization method, *Agricultural Water Management*, 93(3) (2009), 504–510.
- [105] S. Gomez, G. Fuentes, R. Camacho, M. Vasquez, J. M. Otero, A. Mesejo, N. del Castillo, Application of an Evolutionary Algorithm in well test characterization of Naturally Fractured Vuggy Reservoirs, Society of Petroleum Engineering, SPE No. 103931, (2006).
- [106] J.P. Grivet , Nonlinear Population Dynamics in the Chemostat, *Computing in Science and Engineering*, 3: 48-55, IEEE Computer Society. (2001).
- [107] T. Gu , X. Zuo , X. Liu , P. Li, An improved parallel hybrid bi-conjugate gradient method suitable for distributed parallel computing, *Journal of Computational and Applied Mathematics* 226(1) (2009) 55–65.
- [108] B. Guignon, A.M. Ramos, J.A. Infante, J.M. Díaz and P.D. Sanz, Determining thermal parameters in the cooling of a small-scale high pressure freezing vessel. *International Journal of Refrigeration*. 29(7):1152–1159. (2006).
- [109] P. Guttorp. *Stochastic Modeling of Scientific Data*. Oxford Science Publications. Chapman & Hall. (1995).
- [110] J.K. Hale. Ordinary Differential Equations. *Pure and applied mathematics*, Volume XXI, Wiley Interscience, 1969.
- [111] J. Harmand, Rapaport A. and Trofino A., Optimal design of interconnected bioreactors: New results, *AIChE J.*, 49, 2003, 1433–1450, Wiley Subscription Services, Inc., A Wiley Company.
- [112] J. Harmand and Dochain D., The optimal design of two interconnected biochemical reactors revisited, *Computers & Chemical Engineering*, 30(1):70–82. (2005).
- [113] C. Hartman and A. Delgado, Numerical simulation of thermal and fluid dynamical transport effects on a high pressure induced inactivation. *Simulation Modelling Practice and Theory*, vol. 13:109–118. (2005).

-
- [114] C. Hartman, A. Delgado and J. Szymczyk, Convective and diffusive transport effect in a high pressure induced inactivation process of packed food. *Journal of Food Engineering*, 59:33–44. (2003).
- [115] R. Hayashi, Application of High Pressure to Food Processing and Preservation: Philosophy and Development. In *Engineering and Food*, 2:815–826. (1989).
- [116] A.R. Hedar, M. Fukushima, M. Tabu search directed by direct search methods for nonlinear global optimization. *European Journal of Operational Research* 170(2):329–349, (2006).
- [117] F. Herrera, M. Lozano, J.L., Verdegay, Tackling real-coded genetic algorithms: operators and tools for behavioural analysis, *Artif. Intell. Rev.* 12(4) (1998), 265–319.
- [118] D.E. Hertzog, B. Ivorra, B. Mohammadi, O. Bakajin, and J.G. Santiago. Optimization of a microfluidic mixer for studying protein folding kinetics. *Analytical chemistry*, 78(13):4299–4306, (2006).
- [119] D.E. Hertzog, X. Michalet, M. Jäger, X. Kong, J.G. Santiago, S. Weiss, and O. Bakajin. Femtomole mixer for microsecond kinetic studies of protein folding. *Analytical chemistry*, 76(24):7169–7178, (2004).
- [120] K. Hill and Meltz G (1997) Fiber bragg grating technology fundamentals and overview. *Lightwave Technology*, Journal of 15(8):1263–1276
- [121] G.A. Hill and Robinson C. W., Minimum tank volumes for CFST bioreactors in series, *Can. J. Chem. Eng.*, 67, 1989, 818–824, Wiley Subscription Services, Inc., A Wiley Company.
- [122] M.J. Hirsch, C. N. Meneses, P. M. Pardalos, M. G. C. Resende. Global optimization by continuous grasp, *International Journal of Computational Fluid Dynamics*, 1(2):201–212, (2007).
- [123] J. Holland, *Adaptation in natural and artificial systems*, Univ. Michigan Press, (1975).
- [124] R. Horst, P.M. Pardalos, N.V. Thoai. *Introduction to Global Optimization*. Kluwer Academic Publishers. (2000).
- [125] H.S. Horst, R.B.M. Huirne, A.A. Dijkhuizen, "Risk and economic consequences of introducing classical swine fever into the Netherlands by feeding swill to swine", (1999).
- [126] W. Hundsdorfer, J.G. Verwer, Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations, Springer Series in Comput. Math. 33, (2003).
- [127] M. Ibsen, M.K. Durkin, M.J. Cole, R.I. Laming. Sinc-sampled fiber bragg gratings for identical multiple wavelength operation. *IEEE Photonic Technology Letter*. 10(6):842–844. (1998)

- [128] I. Indrawati, A.M. van Loey, C. Smout and M.E. Hendrickx. High hydrostatic pressure technology in food preservation. In *Food preservation techniques*, P. Zeuthen and L. Bogh-Sorensen Eds. Woodhead Publ. Ltd., Cambridge, 428–448. (2003).
- [129] I. Indrawati, L.R. Ludikhuyze, A.M. van Loey and M.E. Hendrickx, Lipxygenase Inactivation in Green Beans (*Phaseolus vulgaris* L.) Due to High Pressure Treatment at Subzero and Elevated Temperatures, *J. Agric. Food Chem.* 48:1850–1859. (2000).
- [130] J.A. Infante, B. Ivorra, A.M. Ramos, and J.M. Rey. On the modeling and simulation of high pressure processes and inactivation of enzymes in food engineering. *Mathematical Models and Methods in Applied Sciences*, 19(12):2203–2229, (2009).
- [131] D. Isebe. Modélisation, simulation numérique et optimisation en génie côtier. *ANRT-Grenoble. Reference: 2007MON20194*, (2008).
- [132] D. Isebe, Azerad, P., Bouchette, F., Ivorra, B., Mohammadi, B. Shape Optimization of Geotextile Tubes for Sandy Beach Protection. *Int. J. Numer. Met. in Eng.* 74(8):1262–1277 (2008)
- [133] R.F. Ismagilov, A.D. Stroock, P.J.A. Kenis, G. Whitesides, and H.A. Stone. Experimental and theoretical scaling laws for transverse diffusive broadening in two-phase laminar flows in microchannels. *Applied Physics letters*, 76(17):2376–2378, (2000).
- [134] B. Ivorra. Optimisation globale semi-déterministe et applications industrielles. *ANRT-Grenoble. Reference: 06/MON2/0061*, (2006).
- [135] B. Ivorra, Mohammadi, B., Dumas, L., Durand, O., Redont, P.: Semi-Deterministic vs. Genetic Algorithms for Global Optimization of Multichannel Optical Filters. *Int. J. of Comput. Science for Eng.* 2(3):170–178 (2006).
- [136] B. Ivorra, B. Mohammadi, and A.M. Ramos (2009) Optimization strategies in credit portfolio management. *Journal Of Global Optimization* 43(2):415–427
- [137] B. Ivorra, Mohammadi, B., Santiago, D.E., Hertzog, J.G.: Semi-deterministic and genetic algorithms for global optimization of microfluidic protein folding devices. *Int. J. Numer. Met. in Eng.* 66(2):319–333 (2006)
- [138] B. Ivorra, A.M. Ramos, and B. Mohammadi (2007) Semideterministic global optimization method: Application to a control problem of the burgers equation. *Journal of Optimization Theory and Applications*, 135(3):549–561.
- [139] M. Jacob, G. Holtermann, D. Perl, J. Reinstein, T. Schindler, M.A. Geeves, and F.X. Schmid. Microsecond folding of the cold shock protein measured by a pressure-jump technique. *Biochemistry*, 38(10):2882–2891, (1999).
- [140] A. Jalvingh, Nielen, M., Maurice, H., Stegeman, A., A.R.W., E. and Dijkhuizen, A. (1999). Spatial and stochastic simulation to evaluate the impact of events and control measures on the 1997-1998. CSF epidemic in The Netherlands, *Prev. Vet. Med.* 42: 271–295.

- [141] A. Jameson, F. Austin, M.J. Rossi, W. Van Nostrand, and Knowles G. Static shape control for adaptive wings. *AIAA Journal*, 32(9):1895–1901, (1994).
- [142] F. Jarre, Kočvara, M., Zowe, J.: Optimal truss design by interior-point methods. *SIAM J. Optim.* 8(4):1084–1107 (1998).
- [143] J.C.L. (2008). Junta de Castilla y Leon - Agricultura y Ganadería, expert opinion elicitation performed for FMD and CSF. Website: <http://www.jcyl.es>
- [144] C.M. Jones, E.R. Henry, Y. Hu, C. Chan, S.D. Luck, A. Bhuyan, H. Roder, J. Hofrichter, and W.A. Eaton. Fast events in protein folding initiated by nanosecond laser photolysis. *Proceedings of the National Academy of Sciences of the United States of America*, 90(24):11860–11864, (1993).
- [145] P. Jung, Baier P, Steil A (1993) Advantages of cdma and spread spectrum techniques over fdma and tdma in cellular mobile radio applications. *Vehicular Technology, IEEE Transactions on* 42(3):357 –364
- [146] Kanno, Y., Ohsaki, M., Murota, K., Katoh, N.: Group Symmetry in Interior-Point Methods for Semidefinite Program. *Optim. and Eng.* 2:293–320 (2001)
- [147] S. Kartsen, Rave, G. and Krieter, J. 2005a. Monte-carlo simulation of CSF epidemics and control I. general concepts and description of the model, *Vet. Microbiol.* 108: 187–198.
- [148] S. Kartsen, Rave, G. and Krieter, J. 2005b. Monte-carlo simulation of CSF epidemics and control II. validation of the model, *Vet. Microbiol.* 108: 199–205.
- [149] S. Kartsen, Rave, G., Teuffert, J. and Krieter, J. (2007). Evaluation of measures for the control of CSF using a simulation model, *Arch. Tierz.* 50: 92–104.
- [150] M. Kavehrad, Zaccarin D (1995) Optical code-division-multiplexed systems based on spectral encoding of noncoherent sources. *Lightwave Technology, Journal of* 13(3):534–545
- [151] A. Kersey (1996) A review of recent developments in fiber optic sensor technology. *Optical Fiber Technology* 2(3):291 – 317
- [152] J. T. Kirby and R. A. Dalrymple. A parabolic equation for the combined refraction diffraction of stokes waves by mildly varying topography. *J. Fluid. Mechanics.*, 136:443–466, (1983).
- [153] J. T. Kirby and R. A. Dalrymple. Combined refraction/diffraction model ref/dif 1, version 2.5. documentation and user’s manual. Research Report No. CACR-94-22, Center for Applied Coastal Research, Department of Civil Engineering, University of Delaware, Newark, (1994).
- [154] J. T. Kirby and R. A. Dalrymple. Refdif1 v2.5 refraction diffraction model. *Technical report, CACR. Short A.D (1999). Beach and Shoreface Morphodynamics.*, Wiley:380pp, (1995).

- [155] T. Kolda, R. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *Siam Review*, 45:385–482, (2003).
- [156] D. Klinkenberg, DeBree, J., Laevens, H. and DeJong, M.C.M. 2002. Within- and between-pen transmission of CSF Virus: a new method to estimate the basic reproduction ratio from transmission experiments, *Epidem. Inf.* 128: 293–299.
- [157] J.B. Knight, A. Vishwanath, J.P. Brody, and R.H. Austin. Hydrodynamic focusing on a silicon chip: Mixing nanoliters in microseconds. *Physical Review Letters*, 80(17):3863–3866, (1998).
- [158] F. Koenen, Van Caenegem, G., Vermeersch, J., Vandenheede, J. and Deluyker, H. 1996. Epidemiological characteristics of an outbreak of Classical Swine Fever in an area of high pig density, *Vet. Record* 139(15): 367–371.
- [159] W. Kowalczyk and A. Delgado, On convection phenomena during high pressure treatment of liquid media, *High Pressure Research* 27(1):85–92 (2007).
- [160] Kyeo-Eun H, Heh Y, Lee S, Mukherjee B, Kim Y (2005) Design and performance evaluation of wdm/tdma-based mac protocol in awg-based wdm-pon. NETWORKING (2005). Lecture Notes in Computer Science, vol 3462, Springer, pp 27–44.
- [161] C. Lam, Yablonovitch E (1995) Multi-wavelength, optical code-division multiplexing based on passive, linear, unitary, filters. In: Signals, Systems, and Electronics, (1995). ISSSE '95, Proceedings., 1995 URSI International Symposium on, pp 299 –302
- [162] L.D. Landau, E. M. Lifshitz (1986) *Theory of Elasticity*. Oxford, England: Butterworth Heinemann.
- [163] C. Lanczos, Solution of Systems of Linear Equations by Minimized Iterations, J. Res. Natl. Bur. Stand. 49 (1952), 33–53.
- [164] M. Larriva, L. Prieto, P. Bruscolini, and A. Rey. A simple simulation model can reproduce the thermodynamic folding intermediate of apoflavodoxin. *Proteins*, 78:73–82, Jan (2010).
- [165] A. Leach. *Molecular Modelling: Principles and Applications*. Prentice Hall, 2 edition, Apr (2001).
- [166] E.W. Lemmon, M.O. McLinden and D.G. Friend, Thermophysical properties of fluid systems. In Linstrom P.J. and Mallard W.G. (Eds.), *NIST Chemistry Web Book. NIST Standard Reference Database*. 69 (June 2005). National Institute of Standards and Technology. Gaithersburg MD, 20899 (<http://webbook.nist.gov>).
- [167] M. A. Lesk. *Introduction to Protein Architecture*. Oxford University Press, Oxford, UK, (2001).
- [168] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, first ed., Cambridge University Press, 2002.

- [169] A.V. Levy, S. Gomez. The tunneling method applied to global optimization. *Numerical Optimization*, 213–244, SIAM, (2008).
- [170] J.G. Lewis, D.G. Payne, R.A. van de Geijn, Matrix-Vector Multiplication and Conjugate Gradient Algorithms on Distributed Memory Computers, Proceedings of Supercomputing’93 (1993) 15–19.
- [171] J.L. Lions, E. Magenes, Problemes aux limites non homogenes et applications - Volume 2, Dunod, 1968.
- [172] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, corrected edition, January (2008).
- [173] L.R. Ludikhuyze, I. van den Broeck, C. A. Weemaes and M. E. Hendrickx, Kinetic Parameters for Pressure–Temperature Inactivation of *Bacillus subtilis* α -Amylase under Dynamic Conditions, *Biotechnol. Prog.*, American Chemical Society and American Institute of Chemical Engineers, 13:617–623. (1997).
- [174] D.G. Luenberger. *Linear and nonlinear programming*. Addison-Wesley, (1984).
- [175] K.C.A.M. Luyben and Tramper J., Optimal design for continuous stirred tank reactors in series using Michaelis-Menten kinetics, *Biotechnol. Bioeng.*, 24, 1982, 1217–1220, Wiley Subscription Services, Inc., A Wiley Company.
- [176] B. Ly–Nguyen, A.M. van Loey, C. Smout, S.E. Özcan, D. Fachin, I. Verlent, S. Vu Truong, T. Duvetter and M.E. Hendrickx, Mild–Heat and High–Pressure Inactivation of Carrot Pectin Methylesterase: A Kinetic Study, *Journal of Food Science*, Institute of Food Technologists. 68(4):1377–1383. (2003).
- [177] Lyytikäinen, T., Niemi, J., Sahlström, L., Virtanen, T. and Lehtonen, H. (2011). The spread of Foot-and-Mouth Disease within Finland and emergency vaccination in case of an epidemic outbreak., *Finnish Food Safety Authority Evira, Helsinki. Research Reports 1/(2011)*. Available at: <http://www.evira.fi/portal/en/evira/publications/?a=viewandproductId=240>
- [178] H. Maaranen, K. Miettinen, A. Penttinen, On initial populations of a genetic algorithm for continuous optimization problems, *J. Glob. Optim.* 37(3) (2007), 405–436.
- [179] B. Malo, Theriault S, Johnson D, Bilodeau F, Albert J, Hill K (1995) Apodised in-fibre bragg grating reflectors photoimprinted using a phase mask. *Electronics Letters* 31(3):223 –225.
- [180] M. Mangen, Nielen, M. and Burrell, A. (2002). Simulated effect of pig-population density on epidemic size and choice of control strategy for Classical Swine Fever epidemics in The Netherlands, *S. Prev. Vet. Med.* 56(2): 141–163.
- [181] E. Mansur, M. Ye, Y. Wang, and Y. Dai. A state-of-the-art review of mixing in microfluidic mixers. *Chinese Journal of Chemical Engineering*, 16(4):503–516, (2008).

- [182] Ministerio de Agricultura, Pesca y Alimentación, Manual práctico de actuaciones contra la PPC. Available at: [http://rasve.mapa.es/Publica/InformacionGeneral/Documentos/Manuales/Manual%20PPC%20enero%20\(2011\).pdf](http://rasve.mapa.es/Publica/InformacionGeneral/Documentos/Manuales/Manual%20PPC%20enero%20(2011).pdf) (2016).
- [183] Y. Marinakis, A. Migdalas, P.M. Pardalos. A new bilevel formulation for the vehicle routing problem and a solution method using a genetic algorithm. *J. Comb. Optim.*, 38(4):555–580, (2007).
- [184] B. Martínez-López. (2009). *Desarrollo de modelos epidemiológicos cuantitativos para el análisis del riesgo de introducción y difusión potencial de los virus de la Fiebre Aftosa y de la Peste Porcina Clásica en España*, PhD thesis, Univ. Complutense de Madrid, Spain.
- [185] B. Martínez-López, Ivorra, B., Ramos, A. and Sánchez-Vizcaíno, J. (2011). A novel spatial and stochastic model to evaluate the within and between farm transmission of CSF Virus: 1. General concepts and description of the model, *Vet. Microbiol.* 147(3): 300–309.
- [186] B. Martínez-López, Ivorra, B., Ramos, A. and Sánchez-Vizcaíno, J. (2012). A novel spatial and stochastic model to evaluate the within and between farm transmission of CSF Virus: 2. Model sensitivity analysis, *Vet. Microbiol.* 155(1): 21–30.
- [187] B. Martínez-López, Pérez, A. and Sánchez-Vizcaíno, J. (2009). A stochastic model to quantify the risk for Classical Swine Fever Virus introduction through import of domestic and wild boars into Spain, *Epidem. Inf.* 137(10): 1505–1515.
- [188] Massey Univ. (2012). InterSpread. Website: <http://www.interspreadplus.com>
- [189] B. Massey and J. Ward-Smith. *Mechanics of Fluids (8th edn)*. Taylor and Francis, (2005).
- [190] MathWorks (2012). Matlab. Website: www.mathworks.com/products/matlab/
- [191] C.C. Mei and D.K.P. Yue. Forward diffraction of stokes waves by a thin wedge. *J. Fluid Mech.*, 99(1):33–52, (1980).
- [192] S. Mehrotra. On the implementation of a primal–dual interior point method. *SIAM Journal on Optimization* 2(4):575–601. (1992).
- [193] C.C. Mei and E.O. Tuck. Forward scattering by long thin bodies. *SIAM J. Appl. Math.*, 39(1):178–191, (1980).
- [194] A. Melinder. *Thermophysical properties of liquid secondary refrigerants*. International Institute of Refrigeration, (1997).
- [195] S. Meulé, Rey V, Certain R, Pinazo C. Wave transformation on a microtidal barred beach (Sète, France). *Journal of Marine System.* (2005). 38:19–34.
- [196] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, 3 edn. Springer, (1998).

-
- [197] K. Mintiens, Laevens, H., Dewulf, J., Boelaert, F., Verloo, D. and Koenen, F. (2003). Risk analysis of the spread of CSF virus through neighborhood infections for different regions in Belgium, *Prev. Vet. Med.* 60(1): 27–36.
- [198] K. Miyagawa and K. Suzuki, Studies on Taka- α -amylase A under high pressure: Some kinetic aspects of pressure inactivation of Taka- α -amylase, *A. Arch. Biochem. Biophys.* 105:297–302. (1964).
- [199] V. Moennig (2000). Introduction to Classical Swine Fever: virus, disease and control policy, *Vet. Microbiol.* 73(2): 93–102.
- [200] B. Mohammadi and O. Pironneau. *Applied shape optimisation for fluids*. Oxford University Press, (2001).
- [201] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Oxford University Press, (2001).
- [202] B. Mohammadi, J-H. Saiac, *Pratique de la simulation numérique*. Dunod, (2002).
- [203] J.J. Moreau. *Mécanique classique*, Volume 1. Masson. (1968).
- [204] J.J. Moreau. *Mécanique classique*, Volume 2. Masson. (1971).
- [205] S.P. Murray, Turbulent Diffusion of Oil in the Ocean, *J. Limnology and Oceanography*, 17(5):651–660 (1972).
- [206] A. Naganathan, U. Doshi, and V. Muñoz. Protein folding kinetics: barrier effects in chemical and thermal denaturation experiments. *Journal of the American Chemical Society*, 129:5673–82, May (2007).
- [207] Navruz I, Guler NF (2007) Optimization of reflection spectra for phase-only sampled fiber bragg gratings. *Optics Communications* 271(1):119 – 123
- [208] J. Necas, W. Jager, J. Stara, O. John. *Partial Differential Equations: Theory and Numerical Solution*. Chapman and Hall. (1999).
- [209] Nelson M. and Sidhu H., Evaluating the performance of a cascade of two bioreactors, *Chemical Engineering Science*, 61, 10:3159–3166 (2006).
- [210] G.L. Nemhauser, Rinnooy Kan, A. H. G., Todd, M. J. (Eds.) *Optimization*. Handbooks in Operations Research and Management Science. 1. Elsevier. (1989)
- [211] N.Q. Ngo, Zheng RT, Ng J, Tjin S, Binh L (2007) Optimization of fiber bragg gratings using a hybrid optimization algorithm. *Lightwave Technology, Journal of* 25(3):799 –802
- [212] Niemi, J., Lehtonen, H., Pietola, K., Lyytikäinen, T. and Raulo, S. (2008). Economic implications of potential Classical Swine Fever outbreaks for Finnish pig production sector., *Prev. Vet. Med.* 84: 194–212.

- [213] T. Norton and D.W. Sun, Recent Advances in the Use of High Pressure as an Effective Processing Technique in the Food Industry. *Food Bioprocess Technol* 1:2–34. (2008).
- [214] Office of response and restoration of the U.S. National Ocean Service, Website: <http://response.restoration.noaa.gov>
- [215] Office of response and restoration of the U.S. National Ocean Service, Incident News Home, Website: <http://www.incidentnews.gov/famous>
- [216] M. Ohsaki. Optimization of geometrically non-linear symmetric systems with coincident critical points. *Int. J. Numer. Met. in Eng.* 48:1345–1357 (2000)
- [217] L. Otero, A.D. Molina–García and P.D. Sanz, Some interrelated thermophysical properties of liquid water and ice I. A user–friendly modeling review for food high–pressure processing, *Critical Reviews in Food Science and Nutrition*, 42(4):339–352. (2002).
- [218] L. Otero, A. Ousegui, B. Guignon, A. Le Bail and P.D. Sanz, Evaluation of the thermophysical properties of tylose gel under pressure in the phase change domain. *Food Hydrocolloids*, 20:449–460. (2006)
- [219] L. Otero, Á.M. Ramos, C. de Elvira and P.D. Sanz, A Model to Design High–Pressure Processes Towards an Uniform Temperature Distribution, *J. Food Eng.* 78:1463–1470 (2007).
- [220] S.H. Park, M.C. Shastry, and H. Roder. Folding dynamics of the b1 domain of protein g explored by ultrarapid mixing. *Nature, Structural Biology*, 6(10):943–947, (1999).
- [221] H.Y. Park, X. Qiu, E. Rhoades, J. Korlach, L. Kwok, W.R. Zipfel, W.W. Webb, and L. Pollack. Achieving uniform mixing in a microfluidic device: Hydrodynamic focusing prior to mixing. *Analytical Chemistry*, 78(13):4465–4473, (2006).
- [222] R. Paskoff. *Côtes en danger*. Masson: Paris. (1993).
- [223] R. Paskoff. *Les plages vont-elles disparaitre?* Le Pommier: Paris. (2006).
- [224] Pille G (2005) Multiplexage optique á répartition par codes spectraux en optique intégrées. *PhD thesis*. University of Montpellier 2 - Montpellier BU Sciences - Reference: TS (2005).MON-228.
- [225] W.L. Price. Global optimization by controlled random search. *Comput. J.* 20:367–370. (1977).
- [226] L. Prieto and A. Rey. Influence of the native topology on the folding barrier for small proteins. *J Chem Phys*, 127:175101, Nov (2007).
- [227] A. Quarteroni. *Numerical Models for Differential Problems*. Springer (2009).

- [228] A. C. Radder. On the parabolic equation method for water-wave propagation. *Journal of Fluid Mechanics*, 95(1):159–176, 1979.
- [229] A.W. Raichle. Numerical predictions of surfing conditions at Mavericks, California. *Shore and Beach*. (1998). 66(2):26–30.
- [230] RAMOGE. La défense des plages contre l'érosion. Guide à l'attention des maitres d'ouvrages, (2002). Available at: www.ramoge.org
- [231] A.M. Ramos, R. Glowinski and J. Periaux, Pointwise control of the Burgers equation and related Nash equilibrium problems: computational approach. *Journal of Optimization Theory and Applications*. 112(3):499–516. (2002).
- [232] R. Ranasinghe and Turner IL. Shoreline response to submerged structures: a review. *Coastal Engineering*. (2006). 536: 65–79.
- [233] J.L. Redondo, J. Fernández, I. García, and P.M. Ortigosa. A robust and efficient global optimization algorithm for planar competitive location problems. *Annals of Operations Research*, 167(1):87–105, (2009).
- [234] J.L. Redondo, J. Fernández, I. García, and P.M. Ortigosa. Solving the Multiple Competitive Location and Design Problem on the Plane. *Evolutionary Computation*, 17(1):21–53, (2009).
- [235] J.L. Redondo, J. Fernández, I. García, and P.M. Ortigosa. Parallel algorithms for continuous competitive location problems. *Optimization Methods and Software*, 23(5):779–791, (2008).
- [236] S. Ribbens, Dewulf, J., Koenen, F., Laevens, H. and de Kruif, A. (2004). Transmission of Classical Swine Fever. A review, *Vet. Q.* 26: 146–55.
- [237] H. Risken. *The Fokker-Planck equation*, volume 18 of *Springer Series in Synergetics*. Springer-Verlag, Berlin, second edition, (1989).
- [238] R. T. Rockafellar. Convex analysis. Princeton Landmarks in Mathematics, Princeton University Press (1997)
- [239] R.T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26:1443–1471. (2002).
- [240] H. Roder. Stepwise helix formation and chain compaction during protein folding. *Proceedings of the National Academy of Sciences of the USA*, 101:1793–1794, (2004).
- [241] J. Rothenberg, Li H, Li Y, Popelek J, Sheng Y, Wang Y, Wilcox R, Zweiback J (2002) Dammann fiber bragg gratings and phase-only sampling for high channel counts. *Photonics Technology Letters, IEEE* 14(9):1309 – 1311
- [242] R. Russell, I.S. Millett, M.W. Tate, L.W. Kwok, B. Nakatani, S.M. Gruner, S.G. Mochrie, V. Pande, S. Doniach, D. Herschlag, and L. Pollack. Rapid compaction during rna folding. *Proceedings of the National Academy of Sciences of the United States of America*, 99(7):4266–4271, (2002).

- [243] A. Saltelli, Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D. Saisana, M., and Tarantola, S., *Global Sensitivity Analysis. The Primer*, John Wiley & Sons. (2008).
- [244] R. Sanson. (1993). *The development of a decision support system for an animal disease emergency*, PhD thesis, Massey University, New Zealand.
- [245] G.A.F. Seber. *Linear regression analysis*. Wiley, New York (1977)
- [246] P. Siarry, G. Berthiau, F. Durbin, J. Haussy. Enhanced simulated annealing for globally minimizing functions of many continuous variables. *newblock ACMTrans. Math. Softw.* 23(2): 209–228, (1997).
- [247] O. Sigmund (2001) A 99 line topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, 21(2):120–127.
- [248] J. Skaar, Risvik KM (1998). A genetic algorithm for the inverse problem in synthesis of fiber gratings. *J Lightwave Technol* 16(10):1928–1932
- [249] G.L.G. Sleijpen and H.A. van der Vorst, Maintaining convergence properties of BiCGstab methods in finite precision arithmetic, *Numerical Algorithms* 10 (1995) 203–223.
- [250] N. D. Socci, J. N. Onuchic, and P. G. Wolynes. Diffusive dynamics of the reaction coordinate for protein folding funnels. *The Journal of Chemical Physics*, 104(15):5860–5868, (1996).
- [251] A. Stegeman, Elbers, A., Bouma, A. and DeJong, M. (2002). Rate of inter-farm transmission of Classical Swine Fever Virus by different types of contact during the 1997-8 epidemic in The Netherlands, *Epidem. Inf.* 128: 285–291.
- [252] A. Stegeman, Elbers, A., Smak, J. and DeJong, M. (1999). Quantification of the transmission of Classical Swine Fever Virus between farms during the 1997-1998. epidemic in The Netherlands., *Prev. Vet. Med.* 42: 219–234.
- [253] K. Suzuki and K. Kitamura, Inactivation of enzyme under high pressure: Studies on the kinetics of inactivation of α -amylase of *Bacillus subtilis* under high pressure., *J. Biochem.* 54(3):214–219. (1963).
- [254] F. Takeshi, Makoto S (2007) A genetic algorithm for the inverse problem in synthesis of fiber gratings. *Toshiba Leading Innovation* 62(11):33–40
- [255] A. Tansakul and P. Chaisawang, Thermophysical properties of coconut milk. *Journal of Food Engineering*, 73(3): 273–280. (2006).
- [256] G. Tremblay, Gillet JN, Sheng Y, Bernier M, Paul-Hus G (2005) Optimizing fiber bragg gratings using a genetic algorithm with fabrication-constraint encoding. *J Lightwave Technol* 23(12):4382

-
- [257] C.W. Tsaoa and D.L. DeVoe. *Lab on a Chip Technology: Volume 1: Fabrication and Microfluidics*, volume 1. Caister Academic Press, (2009).
- [258] J.W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, Reading, MA (1977)
- [259] M.E. Tuckerman. *Statistical mechanics: theory and molecular simulation*. Oxford Graduate Texts. Oxford University Press, Oxford, (2010).
- [260] United States Coast Guards, Spilled Oil Recovery System, Website: http://www.uscg.mil/hq/nsfweb/nsfcc/ops/ResponseSupport/Equipment/sor_sindex.asp, (2010).
- [261] United States Environmental Protection Agency, Oil Spill Response Techniques, Website: [http://www.epa.gov/emergencies/content/learning/oiltech .htm](http://www.epa.gov/emergencies/content/learning/oiltech.htm), (2009).
- [262] H.A. Van der Vorst, Bi-CGSTAB : A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems, *SIAM J. Sci. Stat. Comput.* 13 (1992), 631-644.
- [263] D. Vanderbilt, S.G. Louie. A Monte Carlo simulated annealing approach to optimization over continuous variables. *J. Comput. Phys.* 56: 259–271, (1984).
- [264] F. Verhulst. *Nonlinear differential equations and dynamical systems*. Springer-Verlag., (1990).
- [265] J.G. Verwer, B.P. Sommeijer, W. Hundsdorfer, RKC time-stepping for advection-diffusion-reaction problems, *Journal of Computational Physics* 201 (2004) 61–79.
- [266] A. Viterbi. (1995) *CDMA: Principles of Spread Spectrum Communication*. Addison-Wesley Wireless Communications Serie, Indianapolis.
- [267] V. Voevodin, The Problem of Non-Self-Adjoint Generalization of the Conjugate Gradient Method is Closed, *U.S.S.R. Comput. Maths. and Math. Phys.* 23 (1983) 143–144.
- [268] Warnatz, J. Concentration, Pressure and Temperature dependence of the flame velocity in Hydrogen-Oxygen-Nitrogen Mixtures. *Combustion Science and Technologies.* 26:203–213. (1981).
- [269] D. Wei, Li T, Zhao Y, Jian S (2000) Multiwavelength erbium-doped fiber ring lasers with overlap-written fiber bragg gratings. *Opt Lett* 25(16):1150–1152
- [270] J. Williams, Bennion I, Sugden K, Doran N (1994) Fiber dispersion compensation using a chirped in-fire bragg grating. *Electron Lett* 30:985–987
- [271] E.K. Wilson, Oil Spill’s Size Swells, *Chemical and Engineering News*. American Chemical Society. ISSN 0009-2347, (2010).

-
- [272] M. Wolfson. Validation of population-based disease simulation models: a review of concepts and methods. *BMC Public Health*, 10:710. (2010).
- [273] C.C. Yang. (2008) Optical cdma fiber radio networks using cyclic ternary sequences. *Communications Letters, IEEE* 12(1):41–43
- [274] H. Yang, Cho H, Park J (1999). Optical cdma fiber radio networks using cyclic ternary sequences. *Journal of the Institute of Electronics Engineers of Korea* D 35(4):9–32
- [275] S. Yao and O. Bakajin. Improvements in mixing time and mixing uniformity in devices designed for studies of proteins folding kinetics. *Analytical Chemistry*, 79(1):5753–5759, (2007).
- [276] Q. Ye, A convergence analysis of nonsymmetric Lanczos algorithms, *Math. Comp.* 56 (1991) 677–691.
- [277] H. Yin and Richardson DJ (2009) Springer Berlin Heidelberg
- [278] J. Yoshida, M. Miki, Y. Sakata, Best Combinatorial Crossover in Genetic Algorithms, *Joho Shori Gakkai Kenkyu Hokoku* 2000(85) (2000), 41–44.
- [279] Y. Zhang. Solving Large-Scale Linear Programs by Interior-Point Methods Under the MATLAB Environment. Technical Report TR96-01, Department of Mathematics and Statistics, University of Maryland, Baltimore County, Baltimore, MD, July (1995).
- [280] R. Zwanzig. *Nonequilibrium statistical mechanics*. Oxford University Press, New York, (2001).

Résumé - La modélisation, la simulation et l'optimisation sont trois branches fondamentales et complémentaires des mathématiques appliquées. Elles englobent un large éventail de méthodes et techniques, aussi bien théoriques que numériques, pouvant être utilisées afin de résoudre certains problèmes posés par diverses industries. Durant ces travaux nous en présentons quelques-unes (telles que: modélisation stochastique, simulation numérique à base d'éléments finis et algorithmes heuristiques d'optimisation globale) et les appliquons à la résolution de plusieurs problèmes concrets. En premier lieu, nous nous intéressons à la conception de trois appareils industriels: mélangeur microfluidique (pour l'étude du repliement de protéines), bioréacteur dispersif (pour le traitement d'eaux polluées) et armature (pour le soutien de charges aléatoires). Le deuxième problème étudié porte sur l'analyse de la diffusion et du contrôle par les autorités de la peste porcine classique entre des élevages de porcs dans une région donnée à l'aide d'un modèle hybride Susceptible-Infecté et Individu-Centré. La troisième problématique se centre sur l'évaluation de l'impact sur certains aliments de traitements faisant intervenir de brusques changements de pression et de température dans le but de les stériliser. Finalement, nous présentons un modèle pour l'optimisation de la trajectoire d'un bateau, dit 'Skimmer', dont le rôle est de pomper et de nettoyer de l'eau souillée par un polluant (par exemple, du pétrole) en haute mer.

Mots-clés: *Optimisation globale, Méthodes Méta-heuristiques, Optimisation de forme, Epidémiologie, Traitements alimentaires à Haute-Pression/Température, Nettoyage de pollution en haute Mer, Optimisation structurale.*

Ivorra Benjamin,

Universidad Complutense de Madrid
Facultad de C.C. Matematicas, Depto. de Matematica Aplicada,
Plaza Ciencias 3, 28040 Madrid (España)

email: ivorra@mucm.es / tourbes@hotmail.com